

RESEARCH ARTICLE

Brain Tumour Segmentation Using S-Net and SA-Net

SUNITA ROY¹, RIKAN SAHA¹, SUVARTHI SARKAR², RANJAN MEHERA³,
RAJAT KUMAR PAL¹, (Member, IEEE),
AND SAMIR KUMAR BANDYOPADHYAY⁴, (Senior Member, IEEE)

¹Department of Computer Science and Engineering, University of Calcutta, Kolkata, West Bengal 700106, India

²Department of Computer Science and Engineering, IIT Guwahati, Guwahati, Assam 781039, India

³Anodot Inc., Ashburn, VA 20147, USA

⁴The Bhawanipur Education Society College, Kolkata, West Bengal 700020, India

Corresponding author: Sunita Roy (sunitaroy07@gmail.com)

ABSTRACT Image segmentation is an application area of computer vision and digital image processing that partitions a digital image into multiple image regions or segments. This process involves extracting a set of contours from the input digital image so that pixels belonging to a region share some common characteristics or computed properties, such as color, texture, or intensity. The application domain of image segmentation is widespread and includes video surveillance, object detection, traffic control system, and medical imaging. The application of image segmentation techniques in the field of medical imaging can be further subcategorized into virtual surgery simulation, diagnosis, a study of anatomical structures, measurement of tissue volumes, location of tumours, and other pathologies. In this study, we have proposed two new Convolutional Neural Network (CNN)-based models: (a) S-Net and (b) SA-Net (S-Net with attention mechanism) to perform image segmentation tasks in the field of medical imaging, especially to generate segmentation masks for brain tumours if present in brain Medical Resonance Imaging (MRI) scans. Both proposed models were developed by considering U-Net as the base architecture. The newly proposed models have leveraged the concept of 'Merge Block' to infuse both the local and global context and 'Attention Block' to focus on the region of interest having a specific object. Additionally, it uses techniques, such as data augmentation to utilize the available annotated samples more efficiently. The proposed models achieved a Dice Similarity Coefficient (DSC) measures of 0.78 and 0.81 for the High-Grade Glioma (HGG) and Low-Grade Glioma (LGG) datasets, respectively.

INDEX TERMS Attention block, brain tumour segmentation, convolutional neural network, deep learning, high-grade glioma, low-grade glioma, merge block, U-Net.

I. INTRODUCTION

In the field of image segmentation, the adoption of Deep Learning (DL)-based models [1], [2], [3], [4], [5], [6], [9], [17], [25], [27], [30] is trending upwards, primarily because they do not require a manual feature extraction process and learn the features from the images directly. A large number of application areas have utilized the workflow strategy followed by a DL-based neural network architecture such as object detection and recognition, object

segmentation, object tracking, scene parsing, and medical image diagnosis. Among these application domains, we considered object segmentation as the main domain of this study. The object segmentation process divides an image into different parts carrying different interpretations, such as highlighting the damaged tissue, segmenting the infected or damaged cell, and detecting a specific organ. Different image segmentation techniques used earlier that did not use the concept of the CNN are 1) Threshold-based [36], [37], [38], 2) Edge Detection-based [40], [41], 3) Region-based [42], [43], and 4) Clustering-based [45] methods. On the other hand, CNN-based model architectures

The associate editor coordinating the review of this manuscript and approving it for publication was Essam A. Rashed¹.

were first introduced to perform canonical tasks related to image classification [30] that can classify the whole input image into a single label. In contrast, medical image segmentation requires classifying each pixel of a given input image by extracting pixel-level contextual information. To solve the problem of medical image segmentation tasks using the CNN architecture, some initial attempts were made by Ronneberger et al. [4], who introduced the concept of U-Net (as shown in FIGURE 1), which was developed based on the works of Long et al. [32] using a Fully Convolutional Network (FCN). Furthermore, based on the analysis of some prior studies on neural network-based image segmentation tasks that were implemented on light microscopy images for the ISBI challenge held in 2015, it can be concluded that the U-Net-based model architecture achieves higher performance measures than traditional DL-based image segmentation methods. Moreover, these type of networks become faster as the training processes consider a smaller number of annotated training samples.

U-Net [4], [11], [15], [26] is a state-of-the-art medical image segmentation technique that utilizes the concept of an encoder-decoder architecture. In these type of segmentation models, the concept of skip connections are used to concatenate the low-level, fine-grained features of the encoder or ‘contracting path’ to the high-level, coarse-grained features of the decoder or ‘expansive path’. This concatenation is useful for generating reconstructed fine-grained details of the target segmented mask while performing medical image segmentation tasks. The symmetrical structure of both the ‘contracting’ and ‘expansive paths’ yields a U-shaped architecture. On the other hand, the recreation of spatial information through these concatenations may not provide precise information because the features represented by these down-sampling layers on the ‘contracting path’ are poor and correspond to many lower-level features. These lower-level features may highlight the irrelevant portions of the target segmentation masks; hence, we need to incorporate some attention mechanisms to highlight the relevant portions of the segmentation masks.

On the other hand, attention mechanism-based U-Net models provide a better generalization of the networks, as they always highlight the relevant activations while performing the training. In this manner, we can reduce the requirements of computational resources that are allocated to irrelevant activations. The attention mechanism can be classified into two categories: Hard attention and Soft attention. The hard attention mechanism uses the cropping mechanism to highlight the relevant regions while performing the image segmentation task. This type of attention mechanism works on a particular region at a time and uses a non-differentiable approach; hence, it can not be back-propagated and requires a reinforcement learning approach. On the other hand, the soft attention mechanism uses a weighting approach to assign higher weights to the relevant parts holding the regions of interest and lower weights to the irrelevant parts that mainly cover the background region of the image. In this approach,

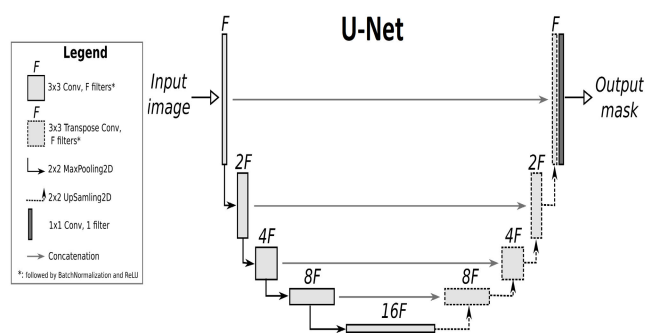


FIGURE 1. Basic U-Net model [4].

we can incorporate the concept of back-propagation while doing the training to learn the appropriate weight vector.

In this study, we implemented segmentation models by considering U-Net as the base architecture that incorporates the following key features to achieve higher performance measures.

Key features of the proposed models are as follows:

- We designed a model by considering a lower number of convolutional layers to define both the down-convolution and up-convolution operations, hence reducing the computational complexity without compromising on the performance measures.
- Introduction of ‘merge block’ (refer to Section III-B) to concatenate feature vectors from all the preceding layers both in the ‘contracting path’ and ‘expansive path’. This layer-wise concatenation increases the model accuracy by accommodating the global and local context of the input images during the training phase.
- The concept of the ‘attention block’ (as shown in FIGURE 9) is also incorporated to focus only on the regions of interest with specific objects like brain tumours.
- Typical DL-based models have a prerequisite of having a large dataset for training purposes. Whereas, the newly proposed models have produced significantly better results by training the model with a sufficiently small number of labeled training samples that were populated by using proper data augmentation techniques.

II. BACKGROUND AND RELATED WORKS

In the literature, there are different types of models such as encoder-decoder-based, pyramid-based, parsing-based, and many more. In our work, we considered an encoder-decoder-based U-Net architecture to solve the problem under consideration. In this section, we classified the U-Net-based image segmentation models (as shown in FIGURE 2 and TABLE 1) into the following primary categories:

The first U-Net was introduced by Ronneberger et al. [4] in 2015, which considered the basic workflow of a Fully Convolutional Network (FCN) and incorporated the concept of skip connection to concatenate the feature maps from

TABLE 1. Details of U-net-based models at a glance.

Sl.	Model	Dataset	Performance Measures													Year	Purpose		
			Avg. IoU	Avg. DSC	Sensitivity	Specificity	Jaccard Dist.	S2S Dist.	F1-Score	AC	AUC	SSIM	PSNR	Avg. Husdorff Dist.					
1	U-Net [4]	PhC-1, U373 (contains 35 partially annotated training images)	0.9203															2015	Biomedical Image Segmentation
		2. DIC-Hela (contains 20 partially annotated training images)	0.7756	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA		
2	3D U-Net [7]	1. Xenopus Kidney Dataset (contains 77 manually annotated training images)	0.723	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2016	Microscopic Kidney Segmentation
3	Attention U-Net [12]	1. CT-150 (contains 150 abdominal 3D CT scans)		0.840±0.087	0.841 ±0.092	0.849 ±0.098		1.920 ±1.284										2018	CT Pancreas Segmentation
		2. TCIA Pancreas-CT Dataset (contains 82 scans)	NA	0.821 ±0.057	0.835 ±0.057	0.815 ±0.093	NA	2.333 ±0.856	NA	NA	NA	NA	NA	NA	NA	NA			
4	Inception U-Net [21]	1. MICCAI BraTS 2017		0.9867													2020	MRI Brain Tumour Segmentation Blood Vessel Segmentation Lung Segmentation	
		2. Retina Image Dataset	NA	0.9582	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA				
		3. CT data from the Kaggle Dataset		0.9857															
5	Residual U-Net [19]	1. Hepatic Veins Dataset (contains 109 cases of abdominal clinical CT volumes)		0.717	0.688	NA	0.561										2019	Hepatic Veins Segmentation Portal Veins Segmentation	
		2. Portal Veins Dataset (contains 109 cases of abdominal clinical CT volumes)	NA	0.765	0.733	NA	0.62	NA	NA	NA	NA	NA	NA	NA	NA				
6	Recurrent U-Net [13]	1a. DRIVE (contains 40 color retinal images)			0.7751	0.9816			0.8155	0.9556	0.9782						2018	Blood Vessel Segmentation Skin Cancer Lesion Segmentation Lung Segmentation	
		1b. STARE (contains 20 color images)			0.8108	0.9871	NA	NA	0.8396	0.9706	0.9909								
		1c. CHASH-DB1 (contains 28 color retina images)			0.7459	0.9836			0.781	0.9622	0.9803								
		2. Skin Cancer Lesion Dataset collected from the Kaggle competition in 2017 (contains 2000 samples in total)	NA	0.8592	0.9334	0.9395	0.938		0.8841	0.938	0.9364	NA	NA	NA	NA				
		3. Kaggle Lung Dataset 2017 (contains 534, 2D samples)		NA	0.9734	0.9866	0.98	NA	0.9638	0.9836	0.9836								
7	Dense U-Net [20]	1. Dataset collected from the Department of Radiology from the University Hospital Brno. (contains 30 MRI brain scans and 15 MRI scans)										0.7855	24.08766			2019	Brain MRI Image Segmentation		
		2. MICCAT 2016 Multiple Sclerosis Segmentation Dataset (contains 30 MRI Brain scans and 15 MRI Scans)	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.7737	24.04422		NA				
8	U-Net++ [14]	1. Cell Nuclei (contains 670 microscopy images)	0.9252														2018	Nuclei Segmentation Polyp Segmentation Liver Segmentation Nodule Segmentation	
		2. Colon polyp (contains 7379 RGB videos)	0.3212																
		3. Liver Dataset (contains 331 CT scans)	0.829	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA				
		4. Lung Nodule (contains 1012 CT scans)	0.7721																
9	Adversarial U-Net [16]	1. DRIVE Dataset (contains 40 color retinal images)	NA	NA	0.7798	0.982	NA	NA	NA	NA	0.9615	NA	NA	NA	NA	2019	Retinal Vessel Segmentation		
10	Cascaded U-Net [18]	1. 3DJRCADb Dataset (contains 20 venous phase enhanced CT volumes)		0.56													2019	Liver Segmentation	
		2. DW-MRI Clinical CT Dataset (contains 100 CT scans)	NA	0.91	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA				
		3. Clinical MRI Dataset (contains MRI for 31 patients)		0.87(Liver),0.6970(Lesion)															
11	Ensemble U-Net [22]	1. BraTS 2017 (contains 285 glioma patients)	NA	NA	0.8113	0.9771	NA	NA	0.8243	0.956	0.9799	NA	NA	NA	NA	2020	Blood Vessel Segmentation		
12	Transformer U-Net [24]	1. TCIA Pancreas Dataset (contains 82 CT scans)		0.7850 ±0.0192													2021	Pancreas Segmentation	
		2. Internal Multi-Organ(IMO) (contains 85 CT scans)	NA	0.8808 ±0.0137	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA				
13	V-Net [8]	1. PROMISE2012 (contains 50 MRI volumes)	NA	0.869 ±0.033	NA	NA	NA	NA	NA	NA	NA	NA	NA	5.71 ±1.20	NA	2016	MRI Prostate Volumes Segmentation		
14	Segnet [10]	1. Camvid (contains 3433 images)	0.601														2017	Road Scene Segmentation Indoor Scene Segmentation	
		2. SUN RGB-D (contains 5285 images)	0.3208	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA				
15	BU-Net [23]	1. BraTS 2017 (contains 285 glioma patients)		0.8920(Whole),0.7830(Core), 0.7360(Enhancing)													2020	Brain Tumour Segmentation	
		2. BraTS 2018 (Contains 285 glioma patients)	NA	0.9010(Whole),0.8370(Core),0.7880(Enhancing)	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA				

low- and high-level feature space to obtain fine-grained output segmented mask.

In 2016, Cicek et al. [7] proposed a 3D U-Net model for volumetric image segmentation. It follows both the semi-automated and fully-automated approaches. In the semi-automated setup, the user annotates some slices of the volume to be segmented that are used to learn the network so that the model could generate a dense 3D segmented output. On the other hand, in the fully-automated approach, the model uses sparsely annotated images as the training samples. In the same year, another model known as V-Net was proposed by Milletari et al. [8] for volumetric image segmentation, which uses an end-to-end CNN model to generate the segmented mask at once. The model also considered a novel objective function based on the dice coefficient metric to address the strong imbalance between the number of foreground and background voxels. The model also uses the concept of data augmentation to cope with a limited amount of annotated training data.

In 2017, Badrinarayanan et al. [10] proposed an encoder-decoder architecture known as SegNet, in which the encoder network follows an identical structure to the VGG16 network. The novelty of SegNet lies in the decoder network that performs non-linear up-sampling and uses the pooling indices computed in the max-pooling step of the corresponding encoder.

In 2018, Oktay et al. [12] proposed an attention-based U-Net that learns how to focus on the target region by suppressing irrelevant parts and highlighting salient features useful for a specific task. After that, Alom et al. [13] designed a recurrent CNN model and a recurrent residual CNN model utilizing the basic U-Net architecture. The use of residual and recurrent residual blocks ensures better segmentation results. In the same year, Zhou et al. [14] proposed the U-Net++ model which is one of the most powerful models for medical image segmentation. The proposed model uses a series of nested dense skip pathways to connect the sub-networks of the encoder and decoder architecture.

In 2019, Feng et al. [18] used a cascaded framework such that the output of one network becomes the input of another network. Here the first blocks of the U-Net architecture extract higher-level feature details, thus highlighting only the areas of interest from the background region, and the successive layers of the U-Net architecture extract lower-level feature details, thereby generating a more accurate target segmentation mask. Kolarik et al. [20] used the concept of a dense block to resolve the vanishing gradient problem owing to the deeper neural network architecture. The network is created by modifying the ResNet architecture. The number of connections between each layer becomes denser as we concatenate the feature maps of all the previous layers. Wu et al. [16] proposed a U-Net-based model that utilizes the Generative Adversarial Network (GAN). The U-GAN model architecture contains a densely connected convolutional network and a novel Attention Gate (AG) in

feature propagation, thereby reducing the number of model parameters. The model also learns to focus on the target structure, without additional supervision. Yu et al. [19] proposed a U-Net architecture based on the ResNet block concept. The model uses a Residual block, which utilizes skip connections to address the vanishing gradient problem caused by a deeper neural network architecture.

In 2020, Zhang et al. [21] proposed a U-Net model utilizing the concept of an inception block that incorporates robustness into the shape and size of the kernels or filters. Concatenation operations are also required to concatenate the feature vector generated by applying different filters. Khoong [22] proposed a U-Net model based on the concept of an ensemble modeling approach that uses multiple diverse base models and predicts an output by reducing the generalization error of the prediction. Rehman et al. [23] proposed a U-Net model known as BU-Net by considering the Residual Extended Skip (RES) and Wide Context (WC) blocks. It also uses a customized loss function to extract more diverse features by increasing the valid receptive field.

In 2021, Petit et al. [24] proposed a model that considers transformer-based modeling techniques used to focus on the global context. By combining the transformer with the U-Net architecture, the performance of the model can be improved by utilizing localized information.

There are many other U-Net variants such as Optimized U-Net [35] and nnU-Net (No New U-Net) [29] for brain tumour segmentation, Swin U-Net [34] for medical image segmentation, D-UNet (Dimension fusion U-Net) [15] for chronic stroke lesion segmentation, and many others.

III. THE PROPOSED MODELS

In this study, we devised different DL-based image segmentation models optimized in terms of the model parameters and performance measures. Based on a detailed literature survey, we learned that the selection of any particular CNN-based model for the task of image segmentation boils down to the trade-off between the number of model parameters and accuracy measures such as Intersection over Union (IoU) or Dice Similarity Coefficient (DSC). By making an exhaustive study of some popular encoder-decoder-based image segmentation models like U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, we have observed that models those are having a smaller number of model weight parameters give poor results in terms of IoU and DSC except for V-Net. On the other hand, if we increase the number of model parameters the model may be over-fitted and also take a lot of time during training, hence making the model slower than the other models. More importantly, CNN-based models that perform medical image diagnosis need to place more emphasis on accuracy than speed during all three stages (training, testing, and validation). In our model, we focused on both the aspects: computation time and accuracy, to design an optimized model for the task of medical image segmentation.

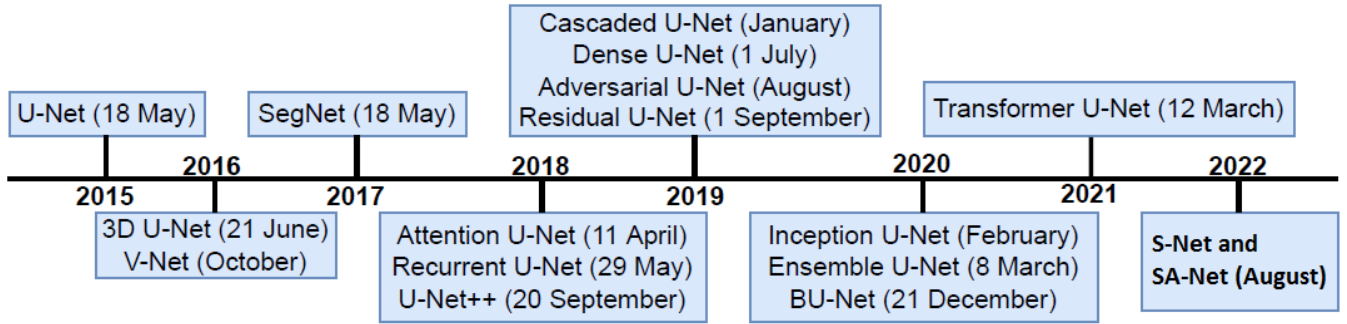


FIGURE 2. Chronological ordering of U-Net and its variants.

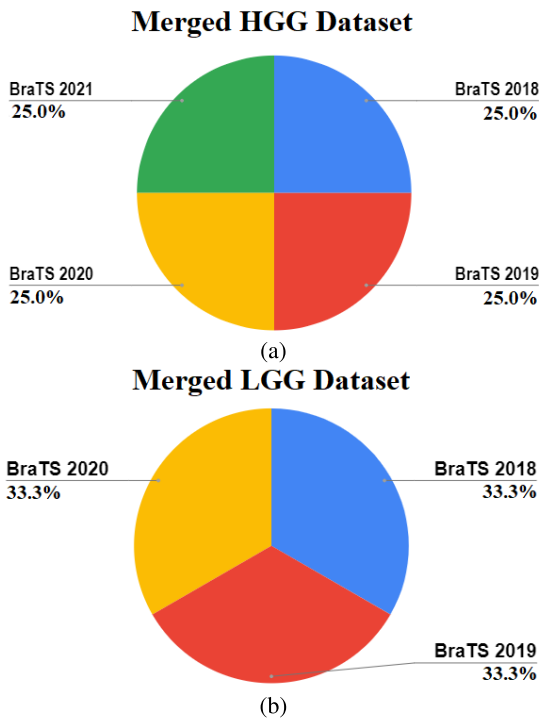


FIGURE 3. (a) Merged HGG dataset details; (b) Merged LGG dataset details.

Both the proposed models were developed by considering U-Net as the base architecture and change the shape of the model asymmetrical in terms of the number of down-convolution and up-convolution layers. Modifications are also made to the CNN architecture by reducing the number of convolutional layers and subsequently decreasing the model weight parameters, which significantly reduced the overall computational overhead. Moreover, the proposed models achieved improved performance measures by incorporating two result-enhancement tools: ‘Merge block’ and ‘Attention block’. In the basic U-Net model, concatenation is performed by considering the low- and high-level features of the same layer. Whereas, in our proposed models, we performed concatenation once after generating all the low- or fine-grained features and high- or coarse-grained features

from all the down-convolution and up-convolution layers, respectively. Furthermore, through this type of concatenation, we can merge both the high-level contextual information of the ‘expansive path’ to the spatial information preserved by the ‘contracting path’ to generate fine-grained details of the target segmentation mask. The merging of spatial information also helps preserve the resolution of the output segmentation mask.

One of the proposed models also utilizes the concept of attention mechanism to assign more weightage to the relevant part of an image other than the irrelevant or background part of the given input image. The weightage factor is defined by a weight vector produced by the ‘attention block’ learned after a suitable training approach. The weight vector carries weights that are higher on the relevant parts as they correspond to the Region Of Interest (ROI) and lower on the irrelevant parts of the input image as they correspond to the background region. Once we obtain the weight vector, we can perform multiplication between the weight vector and the input image to generate a feature vector that can be transformed into a target segmentation mask by applying additional convolutional layers.

A. DATASET CREATION

As is well known, any optimum machine learning model is always driven by the sufficient amount of data samples that are aligned with the task we are performing. Therefore, we must consider the data creation process explicitly and separately. In this study, we used publicly available datasets (BraTS 2018, BraTS 2019, BraTS 2020, and BraTS 2021). The final datasets used by our proposed models were obtained by performing tasks such as Data Acquisition, Data Preprocessing, and Train-Test-Validation Splitting with Data Augmentation. There are different datasets concerning the year but we mainly considered only a few of them. First, we collected the data and preprocessed them to create relevant data points. We then created the train set, test set, and validation set so that they could be propagated to the model. After splitting the data samples, we applied the concept of data augmentation to design models that are much more robust against the variability of tumour shape and size.

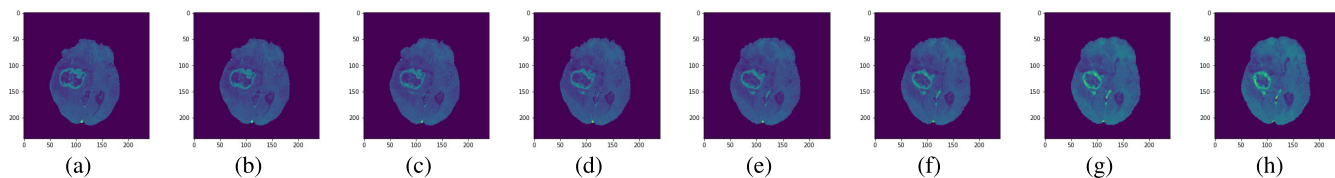


FIGURE 4. (a) – (h) Consecutive 2D brain tumour MRI slices carrying redundant information.

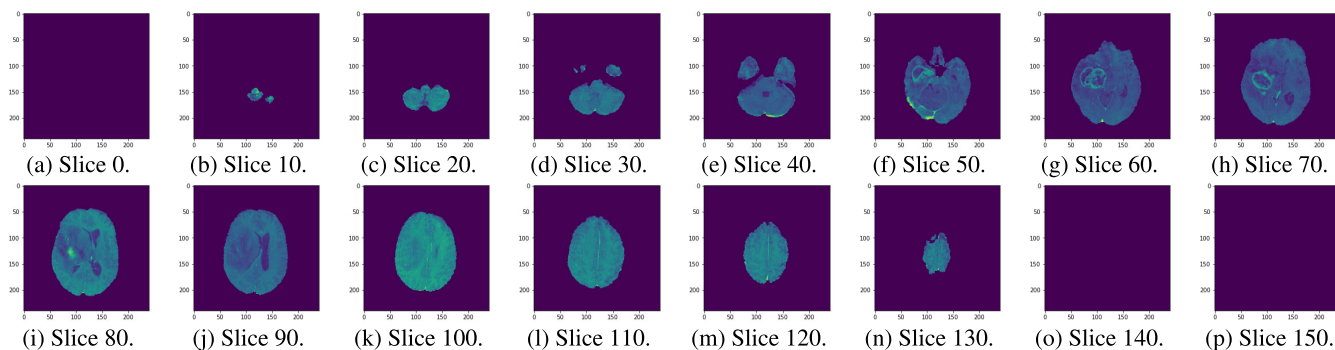


FIGURE 5. (a) – (p) Brain tumour slices with successive gaps of ten.

1) DATA ACQUISITION

We considered four benchmarked brain tumour datasets BraTS 2018, BraTS 2019, BraTS 2020, and BraTS 2021. Each BraTS dataset contains multimodal 3D brain MRIs (T1, T1c, T2, and FLAIR) along with ground truth segmentation masks annotated by medical experts using various MRI scanners from 19 different institutions. Here, each of the segmented mask contains three different sub-regions of a tumour region – Whole Tumour (WT), Tumour Core (TC), and Enhancing Tumour (ET). In our study, we generated a segmentation mask based on the entire tumour region. We downloaded BraTS datasets from an online repository [28]. Each of the dataset contains both the HGG and LGG types of brain tumour images to make the model more robust against variability of data points concerning the medical imaging domain. Gliomas are the primary types of brain tumours and can be classified as HGG [44] and LGG [39]. Furthermore, according to the World Health Organization (WHO), Gliomas can be grouped according to numerical grading structures (Grade I to Grade IV). For example, Grades I and II are grouped as Low-Grade Gliomas or LGG, while Grades III and IV are grouped as High-Grade Gliomas or HGG [31]. We considered two different datasets (HGG and LGG) for the proposed models. First, the HGG dataset was created by taking 50 volumetric data points from these four datasets (as shown in FIGURE 3(a)); hence, the merged HGG dataset contains 200 volumetric multimodal MRIs. On the other hand, each of the 25 images of the merged LGG dataset were taken from three datasets, BraTS 2018, BraTS 2019, and BraTS 2020 (as shown in FIGURE 3(b)), respectively, creating a merged LGG dataset having 75 instances.

2) DATA PREPROCESSING

After gathering all the images, we applied preprocessing techniques to convert these multimodal scans into the corresponding 2D images. Hence, from one single multimodal MRI, we can generate several multiple 2D images. Here, the merged HGG dataset contains 200 volumetric multimodal MRIs, which were further transformed into 1760 2D images before feeding into the proposed DL models. However, the merged LGG dataset contains 75 volumetric instances, which were then converted into 660 2D images. We have also used Python's SimpleITK library to convert the '.nii.gz' format in NumPy arrays. Additionally, each patient's MRI scan volumes were mapped onto the corresponding NumPy array, which was constructed using a tuple of five elements (N, M, S, H, W), where each element can be defined as follows:

N: The total number of HGG/LGG data points considered in the dataset.

M: One of the four modalities (T1, T1c, T2, and FLAIR) along with a segmented mask.

S: Total number of 2D slices corresponding to each MRI 3D volume imagery.

H: Height of the image.

W: Width of the image.

On the other hand, for each patient, a large number of 2D slices were present in the dataset. Moreover, as shown in FIGURE 4, the consecutive 2D slices appear very similar as they correspond to the same patient's MRI data. Hence, if we consider these consecutive 2D slices as our trained data, the model may be over-fitted by these identical image data. FIGURE 4 shows brain tumours MRI slices fetched with a one-slice gap. For example, FIGURE 4(a) shows the 60th brain tumour slice, and FIGURE 4(b) shows

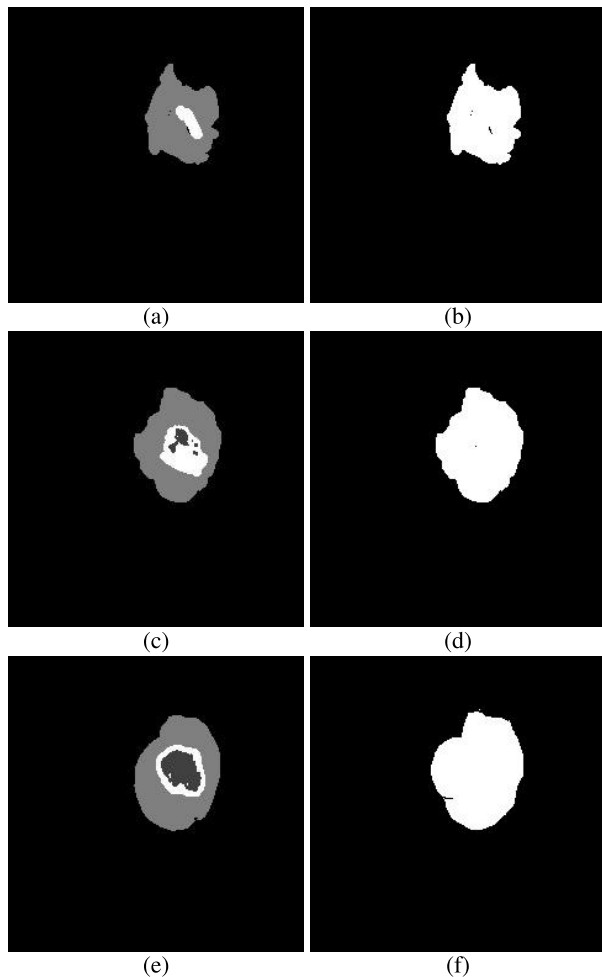


FIGURE 6. (a), (c), and (e) represent multiclass brain tumour segmentation masks; (b), (d), and (f) represent binary segmentation masks after applying threshold-based preprocessing techniques.

the 61st brain tumour slice; hence, both carry almost identical (or redundant) information. On the other hand, FIGURES 4(a) and 4(f) appear different, as they maintain a higher slice gap with each other; hence they can be treated as two different images while creating our training dataset. In FIGURE 5, we show some of the two-dimensional (2D) slices for each 3D volumetric multimodal MRI. Among these slices, some are relevant and some are irrelevant as depending on the coverage type (fully or partially) of the entire brain region. Moreover, as shown in FIGURES 5(a)-5(f) and FIGURES 5(l)-5(p), the slices that are captured near the left or right boundary of the skull do not adequately cover the entire brain region. To resolve this problem, we considered only some of the middle slices, as shown in FIGURES 5(g)-5(k), which are also non-consecutive. In our dataset, we started the data collection from slice number 60 among 155 slices. Furthermore, we have taken the slices five places apart; for example, if the current 2D brain image is taken from a slice number 60, then the next 2D image will be taken from the slice number 65 to decrease the redundancy.

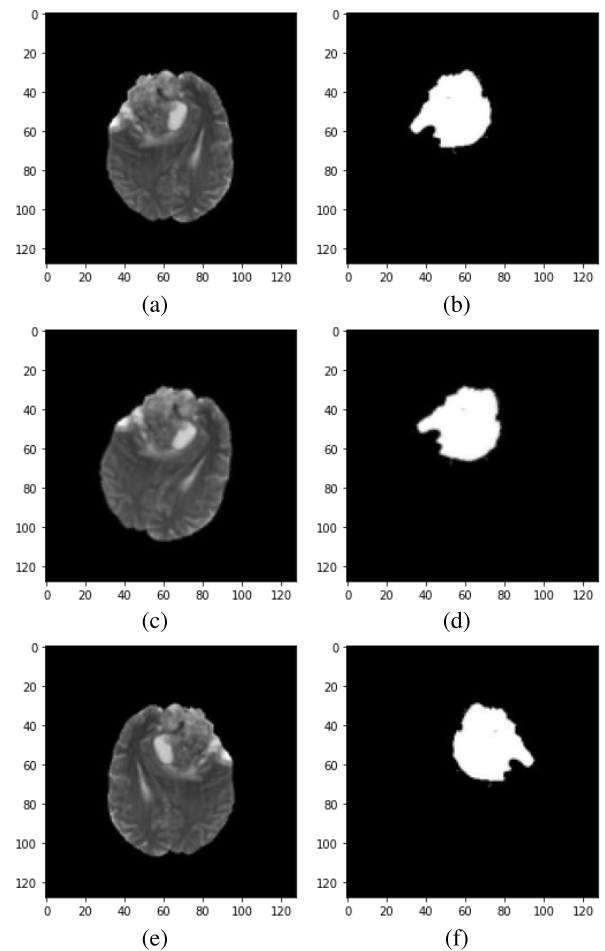


FIGURE 7. (a) and (b) represent the image and its corresponding segmentation mask without image transformation, respectively; (c) and (d) represent the image and its corresponding segmentation mask with rotation, respectively; (e) and (f) represent the image and its corresponding segmentation mask with horizontal flipping, respectively.

3) TRAIN-TEST-VALIDATION SPLITTING WITH DATA AUGMENTATION

As we are designing a binary segmentation model, our segmentation mask contains two regions: the tumour (whole tumour) and the background region as shown in FIGURE 6. Moreover, we need to apply a threshold-based preprocessing technique to reduce the number of classes in the original segmented mask and make them suitable for binary segmentation tasks. Furthermore, we must evaluate model performance on unseen data when exploring various model architectures. For example, in a predictive model, we train the model on limited data and evaluate it on unseen data. We divided the total dataset into a train set, test set, and validation set. In our proposed model, we created a training dataset from a merged dataset containing four benchmark datasets BraTS 2018, BraTS 2019, BraTS 2020, and BraTS 2021. As shown in FIGURE 8, we divided the merged image dataset into an 80% train set, 10% test set, and 10% validation set. Subsequently, we applied the concept of data augmentation to deal with the problem of limited data availability while doing the training.

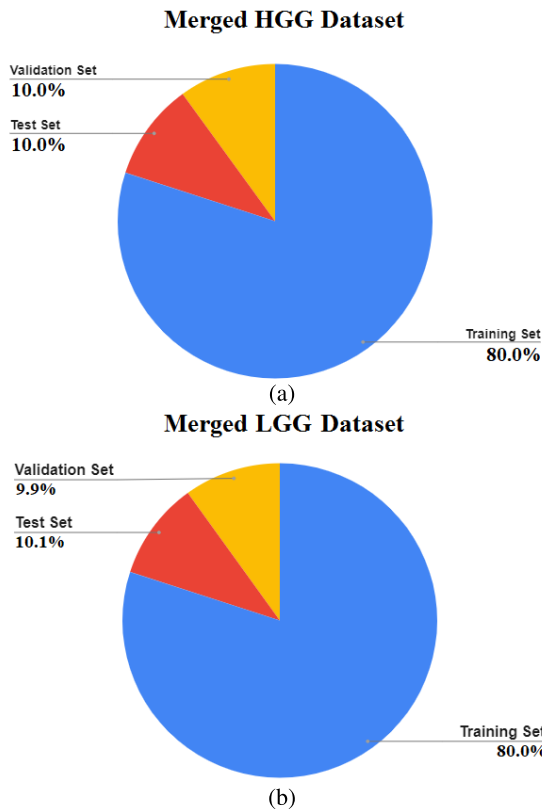


FIGURE 8. The train-test-validation splitting ratio on (a) Merged HGG dataset; and (b) Merged LGG dataset.

In medical imaging, it is very difficult to collect a large amount of data owing to ethical considerations followed by medical practitioners. In addition, the data annotation process takes enormous time and effort for medical practitioners, thus making the process more complicated and expensive. Therefore, data augmentation techniques are required to make the model more robust against the variability of the images in terms of the shape, size, and the relative position of the tumour region. Hence, we should populate our dataset with various images generated by applying image transformation techniques like random rotation, flipping, shear, translation, and so on. In our experiment, we have considered two types of image transformation techniques: random rotation in the range of 40° and horizontal flipping around the y-axis.

a: RANDOM ROTATION

While performing image transformation using the random rotation technique, we need to define an angle that can be limited to 360° . In our experiment, we limited the range to 40° to increase the number of images in all the data sources.

b: HORIZONTAL FLIPPING

Image flipping can be performed along the x- or y-axes, depending on whether vertical or horizontal flipping is performed. In the medical image analysis technique, the

TABLE 2. Number of images both before and after applying the data augmentation for the HGG dataset.

	TRAIN SET	TEST SET	VALIDATION SET
BEFORE	1760	220	220
AFTER	5280	660	660

TABLE 3. Number of images both before and after applying the data augmentation for the LGG dataset.

	TRAIN SET	TEST SET	VALIDATION SET
BEFORE	660	83	82
AFTER	1980	249	246

image cannot be flipped vertically, resulting in upside-down images. Hence, we applied horizontal flipping to generate the left and right views of a brain tumour image. After applying the above two image transformation techniques, we increased the number of images in both the HGG and LGG datasets, as shown in TABLE 2 and 3, respectively. FIGURE 7 also shows the resultant images generated after applying various image transformation techniques.

B. PROPOSED S-NET AND SA-NET MODEL ARCHITECTURES

Here, we will describe the two proposed models in detail. Section B.1 covers the architectural details, where we first describe the architecture of the S-Net model and, then explain the additional blocks that we incorporated to convert it into the SA-Net model architecture. Section B.2 illustrates all the mathematical formulations behind all the operations applied to our proposed model architectures, and Section B.3 has covered the hyperparameter tuning process applied while training the model to achieve good segmentation results w.r.t. various performance measures, such as IoU, and Dice Coefficient.

1) DEFINING MODEL ARCHITECTURE

The network architectures of our proposed models consider the basic U-Net as the baseline architecture and redefine the structure of the ‘contracting’ and ‘expansive path’ to reduce the computational overhead. As the original U-Net structure, our models also consist of two main paths: i) the ‘contracting path’ that encodes the whole image and ii) the ‘expansive path’ that recovers the original resolution. Both encoding and decoding activities were performed using five levels of down-sampling and four levels of up-sampling computations, respectively. At each level, the model utilizes the concept of residual blocks considering only one convolutional layer instead of two, thereby reducing the number of model weight parameters. TABLE 4 shows the layer-wise feature map details along with the related operations.

While defining the convolutional layer, we used a 3×3 kernel along with the ReLU activation function. The models applied five convolutional layers with different feature sizes (64, 128, 256, 512, and 1024), as shown in

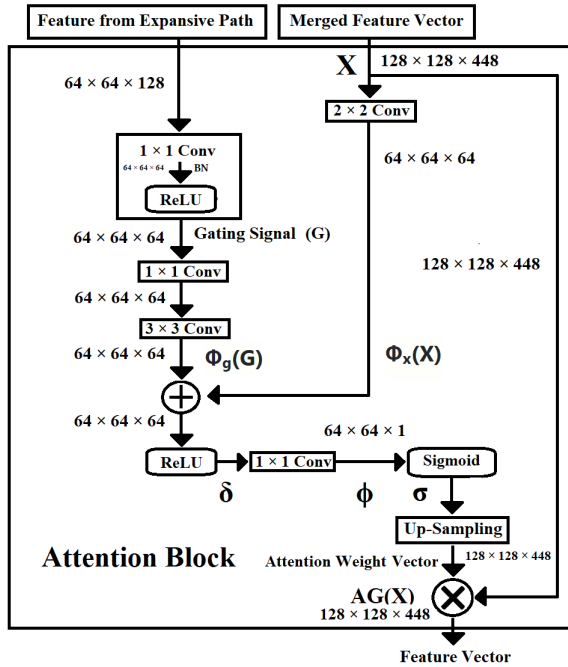


FIGURE 9. Diagram of attention block used in the proposed SA-Net.

FIGURES 10 and 11. Each convolutional layer is followed by a pooling layer. We added two dropout layers with a dropout rate of 0.2 at the end of the encoder stage and the beginning of the decoder stage, respectively, to reduce the problem of over-fitting. On the decoder side, we applied four deconvolutional layers with different feature sizes (512, 256, 128, and 64). While applying the deconvolution operation, an up-sampling layer is used to increase the dimension of the feature maps so that we can get back the segmented image having the same dimension as the input image. Unlike the original U-Net model, we did not merge the feature maps level-by-level; here, merging was performed using a ‘merge block’ once before applying the attention mechanism. Furthermore, before merging, we equalized the dimension of all feature maps generated by all convolutional and deconvolutional layers.

Finally, after merging, the proposed SA-Net model uses the concept of the ‘attention block’ (as shown in FIGURE 9) before the final layer learns the target structure by highlighting salient features of interest and suppressing irrelevant regions not beneficial for a specific task. We have the output feature maps of the ‘merge block’ and the final up-sampled feature of the ‘expansive path’ as the two inputs of the ‘attention block’. The feature representation from the ‘expansive path’ is transformed into a gating signal using a convolutional layer and a Rectified Linear Unit (ReLU) activation function. The gating signal and merged feature vector of the ‘merge block’ are passed through convolutional layers to perform feature addition. The output feature vector is then transformed into a feature vector using the ReLU and sigmoid activation functions. Finally, we do the up-sampling

to transform the feature vector to the corresponding weight vector, which has the same resolution as the merged feature vector. The final feature vector is generated by performing a multiplication between the weighted feature vector and the merged feature vector to generate a feature vector carrying the weighted feature representation of the target segmentation mask. Finally, we applied additional convolutional layers to obtain the target binary segmentation mask.

2) MATHEMATICAL FORMULATION FOR DIFFERENT OPERATIONS

The proposed models use the basic workflow followed by the U-Net architecture, which consists of two paths: a ‘contracting path’ and an ‘expansive path’. Furthermore, the structural details were changed to create a lightweight model that has a smaller number of model parameters than the basic U-Net model and can achieve better performance measures. In the ‘contracting path’, we are designing the down-convolution operation by using a sequence of convolutional layers with a ReLU activation function followed by a max-pooling layer. In the ‘expansive path’, we design an up-convolution operation using a sequence of up-sampling layers. After the completion of successful feature extraction from both the down-sampling and up-sampling layers, we need to merge the features that involve another two sub-operations namely feature equalization and feature concatenation. Finally, we designed an ‘attention block’ that uses two operations: The generation of the weight vector and the attention function, to obtain the target segmentation mask as shown in FIGURE 9.

a: CONTRACTING PATH

Here the ‘contracting path’ consists of repeated application of only one convolutional layer instead of two as the basic U-Net model. Each CNN layer is followed by a non-linear activation function called ReLU and a down-sampling module that consists of a max-pooling operation with a stride of 2. After each down-sampling operation, the number of feature channels get doubled.

i. DOWN-CONVOLUTION

While defining the down-convolution operation, we must define the number of filters along with their size. We must mention the padding related information along with the activation function. In our proposed models, we used different numbers of filters (e.g., 64, 128, 256, 512, and 1024) where the size of the filters remained fixed (e.g., 3×3).

$$[n_h, n_w, n_c] * [f, f, n_c] \Rightarrow [(n_h + 2P - f + 1), (n_w + 2P - f + 1), n_f] \quad (1)$$

where

n_h : Height of the image,

n_w : Width of the image,

f : Filter size,

n_c : Number of channels in the image,

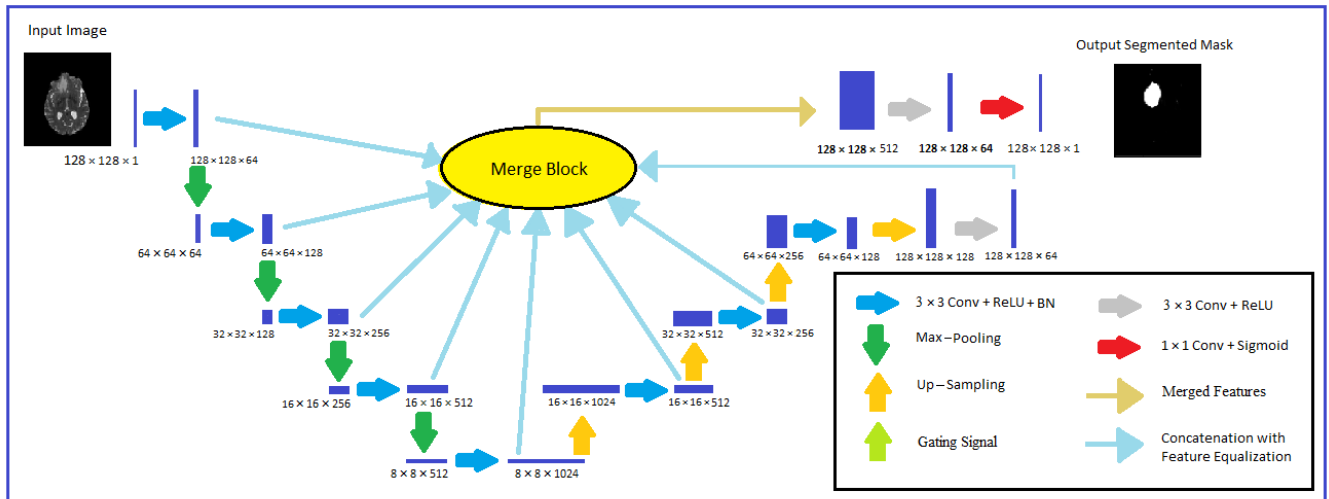


FIGURE 10. The proposed S-Net model architecture.

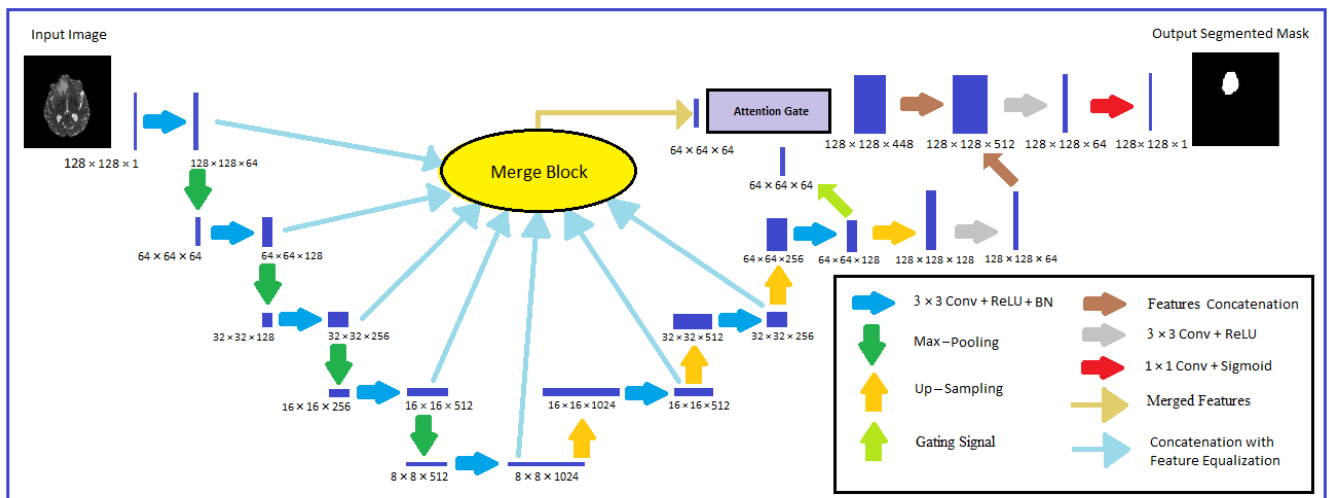


FIGURE 11. The proposed SA-Net model architecture.

P : Used padding,
 n_f : Number of filters,
 $*$: Convolution operation.

ii. ReLU ACTIVATION FUNCTION

This is a non-linear activation function that outputs the input directly if it is positive; otherwise, it outputs zero. It is a very commonly used activation function in neural networks, particularly in Convolutional Neural Networks (CNNs) and multilayer perceptrons. We used this activation function in all convolutional layers present in both the ‘contracting path’ and ‘expansive path’ except the last layer or the final layer.

$$\sigma(z) = \max(z, 0) \tag{2}$$

where σ : Activation function.

iii. MAX-POOLING

The type of pooling operation used in our proposed models is max-pooling. While defining the max-pooling operation we need to define the stride length. In our proposed models, we used a stride length of 2 to perform the pooling operation. For a feature map with dimensions $[n_h, n_w, n_c]$, the dimensions of the output feature map obtained after a pooling operation are calculated as follows:

$$\left\lfloor \frac{(n_h + 2P - f)}{s} + 1 \right\rfloor, \left\lfloor \frac{(n_w + 2P - f)}{s} + 1 \right\rfloor, n_f \tag{3}$$

where s : Stride length.

b: EXPANSIVE PATH

The ‘expansive path’ consists of an up-convolution operation that halves the number of feature channels and is defined by one 3×3 convolutional layer.

i. UP-CONVOLUTION

While defining the up-sampling or up-convolution operation we need to define the up-sampling factors for the rows and columns. In our proposed models, we used an up-sampling factor of 2 for both rows and columns. For a feature map with dimensions $[n_h, n_w, n_c]$, the dimensions of the output feature map obtained after an up-sampling operation are calculated as follows:

$$[n_h \times u_r, n_w \times u_c, n_c] \quad (4)$$

where

u_r : Up-sampling factors for rows,

u_c : Up-sampling factors for columns.

c: MERGE BLOCK

The merging operator is used to concatenate the feature maps of both the ‘contracting path’ and ‘expansive path’. Here merging operator is applied simultaneously and requires some down-sampling and up-sampling operations to equalize the feature vector before performing the final concatenation.

i. FEATURE EQUALIZATION

For a feature map with dimensions $[n_{h1}, n_{w1}, n_{c1}]$, if we want to convert the input feature map to the output feature map of dimensions $[n_{h2}, n_{w2}, n_{c2}]$ we need to incorporate up-convolution and down-convolution layers to transform the input feature map to the desired output feature map. The number of down-convolution and up-convolution layers will be determined by the following rules as shown in Equations (5), (6), (7), and (8), respectively. For simplicity we assume that the images are square in size; hence $h_1 = w_1$ and $h_2 = w_2$. Here, we formulate the equations by considering only one form as follows:

$$\begin{aligned} &\text{If } c_1 > c_2 \text{ then} \\ &a = c_1/c_2 \\ &\text{else} \\ &a = c_2/c_1 \end{aligned} \quad (5)$$

Now the number of down-convolution layers is determined by a factor x such that:

$$2x = a \quad (6)$$

If $h_1 > h_2$ then

$$b = h_1/h_2$$

else

$$b = h_2/h_1 \quad (7)$$

Now the number of up-convolution layers is determined by a factor y such that:

$$2y - 1 = b \quad (8)$$

ii. FEATURE CONCATENATION

If we want to concatenate the feature map of dimensions $[n_{h1}, n_{w1}, n_{c1}]$ with a feature map of dimensions $[n_{h2}, n_{w2}, n_{c2}]$, the output feature map can be calculated using the following formula as shown in Equation (9). In the concatenation operation, the height and width of the two feature maps should be identical, hence, we assume that $h_1 = h_2 = h$ and $w_1 = w_2 = w$. Thus, the dimension of the output feature map can be calculated as follows:

$$[n_h, n_w, n_{(c_1+c_2)}] \quad (9)$$

where $n_{(c_1+c_2)}$: Merged channel.

d: ATTENTION BLOCK

After doing the merging operation, the merged feature vector is applied to the ‘attention block’ along with the feature vector generated by the ‘expansive path’. The output of the ‘attention block’ is passed through two convolutional layers to obtain the final output feature map with the same dimension as the input image so that we can successfully generate the output segmentation mask.

i. SIGMOID ACTIVATION FUNCTION

The sigmoid activation function is a special form of the logistic function and is typically denoted by $\sigma(x)$ or $\text{sig}(x)$. Sigmoid activation function most often shows a return value in the range of 0 and 1. A sigmoid activation function was used after the last convolution operation. We also use this function in the ‘attention block’. This is given by Equation (10):

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (10)$$

ii. ATTENTION FUNCTION

Given an intermediate feature map $X \in \mathbb{R}^{C_1 \times H_1 \times W_1}$ and gating signal $G \in \mathbb{R}^{C_2 \times H_2 \times W_2}$, with C_1 or C_2 channels and feature maps of size $H_1 \times W_1$ or $H_2 \times W_2$, we need to perform some linear mapping to generate a feature map X' in the $\mathbb{R}^{C_1 \times H_1 \times W_1}$ dimensional space as shown in Equations (11) and (12).

$$X' = AG(X) \otimes X \quad (11)$$

where

\otimes denotes element-wise multiplication, \oplus denotes element-wise addition, and $AG(X) \in \mathbb{R}^{C_1 \times H_1 \times W_1}$ is a three-dimensional weight map generated by the corresponding Attention Gate (AG) module. Here, the output of the Attention Gate module depends on the whole feature map X .

$$AG(X) = \sigma(\phi(\delta(\varphi_x(X) \oplus \varphi_g(G)))) \quad (12)$$

where

σ : Sigmoid activation function,

δ : ReLU activation function,

ϕ : Linear transformation implemented as 1×1 convolution,

φ_x : Linear transformation implemented on input X,
 φ_g : Linear transformation implemented on gating signal G.

3) HYPERPARAMETERS OF THE MODEL

Hyperparameter tuning is a critical task when designing a CNN model. However, tuning measures may not follow a standard format; instead, we get measures that depend on different factors such as the type of dataset we use and, the type of task we are performing.

- The learning rate, momentum, and decay values are set to 0.001, 0.9, and 0.0005, respectively. These parameters were fixed throughout the training process.
- We have considered two different types of optimizers: Stochastic Gradient Descent (SGD) and Adam. In comparison, SGD outperformed Adam.
- ReLU is used as an activation function for all hidden layers, while the sigmoid activation function is used by the last layer only, as we are performing a binary segmentation task.
- While tuning these parameters, we considered the most popular loss functions suitable for semantic segmentation tasks. Among these loss functions, Binary Cross Entropy (BCE) provides better performance measures owing to the nature of binary segmentation tasks at hand.
- The proposed models give improved performance measures with a dropout rate of 0.2 among all the other values, such as 0.5 and 0.1, respectively. We applied a standard dropout [33], which sets the dropout rate to 0.2 for the middle layers.
- The proposed models were trained for epoch values 20, 50, 75, and 100.
- The kernel size and pool size are defined as 3×3 and 2×2 , respectively, to make the models suitable for the application under consideration.
- The proposed models were tested with batch sizes 16, 32, and 64. According to our observation, the increase in batch size has a direct impact on the consumption of the resources in terms of both CPU and memory requirements. We experimented by keeping the batch size fixed at 16.

IV. EXPERIMENTAL DETAILS

All implementation details, including the system specification, performance measures, and result analysis, are explained in the following subsections.

A. SYSTEM SPECIFICATION

We used the Google Colab Pro+ environment, which allows us to write and execute Python code designed for a specific machine learning-based application through the browser. Furthermore, no initial setup is required because it has a built-in Jupyter notebook service that requires no setup while providing resources, including Graphics Processing Units (GPUs). The experimental environment consisted of

a Windows 10 operating system, Intel® Xeon® 2.30 GHz Processors, 26 GB running memory (RAM), NVIDIA Tesla P100-PCIE-16 GB GPU, and 129 GB disk space.

B. EVALUATION METRICS

To evaluate the performance of the proposed S-Net and SA-Net models, we considered the Intersection over Union (IoU), Dice Similarity Coefficient (DSC), Sensitivity, Specificity, and Accuracy as the figures of merit. These metrics can be formulated using TP, FP, TN, and FN, which are the abbreviations of the number of instances for true positives, false positives, true negatives, and false negatives, respectively. The metrics are defined as follows:

1) INTERSECTION OVER UNION (IoU)

This is calculated by considering the area of overlap between the predicted segmentation result and ground truth result divided by the area of union between the predicted mask and ground truth mask.

$$IoU = \frac{target \cap prediction}{target \cup prediction} \quad (13)$$

2) DICE SIMILARITY COEFFICIENT (DSC)

This is also a calculation of overlap-based metric to measure the spatial overlap between the ground truth mask and predicted mask.

$$DSC = \frac{2 \times (target \cap prediction)}{target + prediction} \quad (14)$$

3) SENSITIVITY

It is the percentage of actual positive values correctly identified.

$$Sensitivity = \frac{TP}{TP + FN} \quad (15)$$

4) SPECIFICITY

It is the percentage of actual negative values correctly identified.

$$Specificity = \frac{TN}{TN + FP} \quad (16)$$

5) ACCURACY

It is a probabilistic metric that measures the degree to which the segmentation results agree with the ground truth segmentation mask.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

C. RESULTS AND ANALYSIS

In this section, we compare our proposed models with other well-known models such as U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively. We considered only the U-Net-based models as these are effectively and efficiently used in the field of

TABLE 4. Layer-wise feature map details along with the related operations.

Input	Feature Size	Two-Dimensional Operation
Input Image	$128 \times 128 \times 1$	No operation performed yet
Convolution Layer 1	$128 \times 128 \times 64$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Max-Pooling	$64 \times 64 \times 64$	Reduce feature map using a stride of 2
Convolution Layer 2	$64 \times 64 \times 128$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Max-Pooling	$32 \times 32 \times 128$	Reduce feature map using a stride of 2
Convolution Layer 3	$32 \times 32 \times 256$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Max-Pooling	$16 \times 16 \times 256$	Reduce feature map using a stride of 2
Convolution Layer 4	$16 \times 16 \times 512$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Dropout	$16 \times 16 \times 512$	Apply a dropout rate of 0.2
Max-Pooling	$8 \times 8 \times 512$	Reduce feature map using a stride of 2
Convolution Layer 5	$8 \times 8 \times 1024$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Dropout	$8 \times 8 \times 1024$	Apply a dropout rate of 0.2
Up-Sampling	$16 \times 16 \times 1024$	Increase the feature map by factor 2
Convolution Layer 6	$16 \times 16 \times 512$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Up-Sampling	$32 \times 32 \times 512$	Increase the feature map by factor 2
Convolution Layer 7	$32 \times 32 \times 256$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Up-Sampling	$64 \times 64 \times 256$	Increase the feature map by factor 2
Convolution Layer 8	$64 \times 64 \times 128$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Up-Sampling	$128 \times 128 \times 128$	Increase the feature map by factor 2
Convolution Layer 9	$128 \times 128 \times 64$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU} + \text{BN})$
Gating Signal	$64 \times 64 \times 64$	Resize the down-layer feature map into the same dimension as the up layer feature map using $1 \times 1 \text{ Conv}$
Merge Block	$64 \times 64 \times 64$ for SA-Net $128 \times 128 \times 512$ for S-Net	Merge all the feature maps
Attention Block	$128 \times 128 \times 448$	Apply attention mechanism on both the merged feature and gated output
Feature Concatenation	$128 \times 128 \times 512$	Concatenate the merged feature maps
Convolution Layer 10	$128 \times 128 \times 64$	$1 \times (3 \times 3 \text{ Conv} + \text{ReLU})$
Final Convolution Layer	$128 \times 128 \times 1$	$(1 \times 1 \text{ Conv} + \text{Sigmoid})$

medical image segmentation tasks and have become state-of-the-art DL-based image segmentation models by providing consistent and good performance measures. In TABLE 5, we have compared the models based on performance metrics such as IoU, DSC, Sensitivity, Specificity, and Accuracy, respectively. We run the experiment for four epochs: 20, 50, 75, and 100; and have recorded our observations accordingly.

In FIGURES 12-22, we considered the best performance measures by taking the maximum metric value over all epochs. FIGURES 12-16 show the performance measures observed on the HGG dataset and FIGURES 17–21 show the performance measures observed on the LGG dataset. FIGURE 22 shows the number of model weight parameters for each model under consideration. Moreover, FIGURE 23 shows how the performance of the SA-Net model improved over epochs during the training phase.

We concluded the following observations based on the results recorded in TABLE 5. The model-wise predicted segmentation masks are also recorded in TABLE 6, considering the HGG and LGG datasets separately.

- 1) Segmentation models with a larger number of model weight parameters provide better segmentation results in terms of IoU and DSC compared to models with fewer model weight parameters.
 - a) In this section, we highlight the behavior of the models based on the IoU scores observed on the HGG dataset as shown in FIGURE 12.
 - Models such as U-Net, Attention U-Net, Attention ResU-Net, SegNet, and V-Net are

considering larger model weight parameters and are providing better IoU measures of 77.38, 77.46, 77.47, 76.46, and 76.54, respectively.

- Models such as U-Net++ and BU-Net are considering smaller model weight parameters, yield poor performance measures in terms of IoU measures of 65.27 and 73.21, respectively.
 - On the other hand, our two proposed models, S-Net and SA-Net, give comparatively higher IoU values of 77.36 and 77.61, respectively, despite considering the smaller model weight parameters.
- b) In this section, we highlight the behavior of the models based on the IoU scores observed on the LGG dataset as shown in FIGURE 17.
 - Models such as U-Net, Attention U-Net, Attention ResU-Net, SegNet, and V-Net are considering larger model weight parameters as shown in FIGURE 22, and providing better IoU measures of 80.66, 80.74, 80.41, 79.33, and 80.32, respectively.
 - Models such as U-Net++ and BU-Net are considering smaller model weight parameters, provide poor performance measures in terms of IoU measures of 70.61 and 74.89, respectively.
 - On the other hand, our two proposed models, S-Net and SA-Net, give comparatively higher

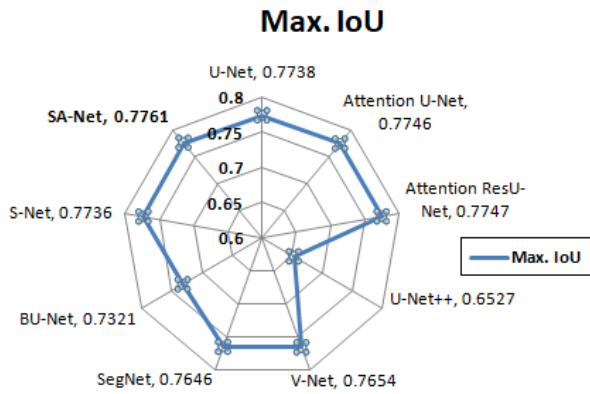


FIGURE 12. Model-wise comparison for IoU observed on HGG dataset.

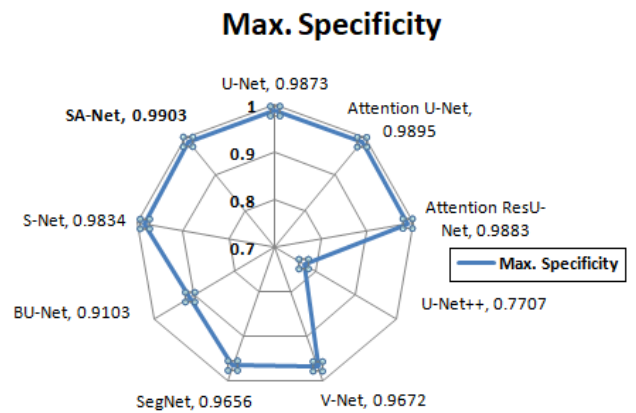


FIGURE 15. Model-wise comparison for Specificity observed on HGG dataset.

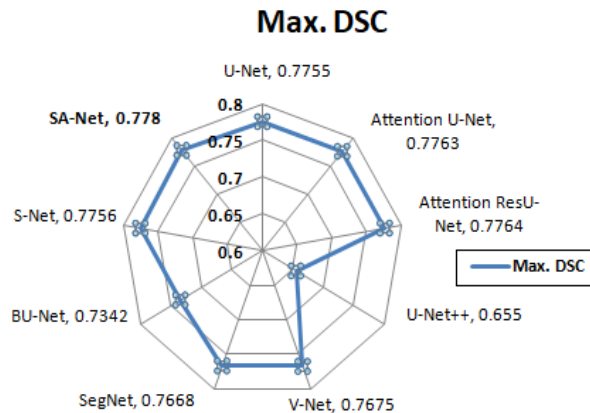


FIGURE 13. Model-wise comparison for DSC observed on HGG dataset.

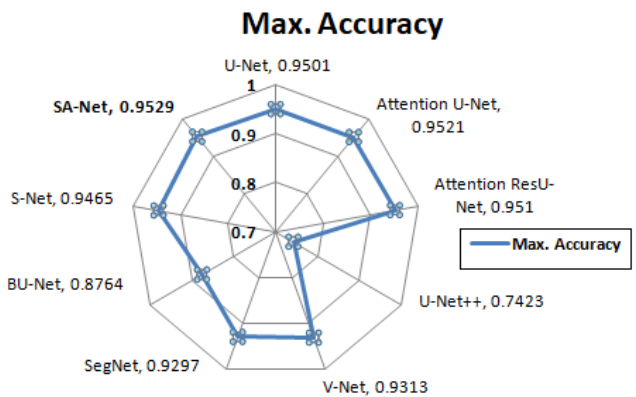


FIGURE 16. Model-wise comparison for Accuracy observed on HGG dataset.

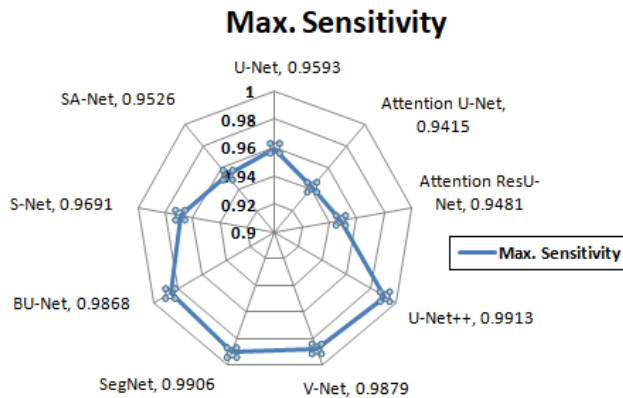


FIGURE 14. Model-wise comparison for Sensitivity observed on HGG dataset.

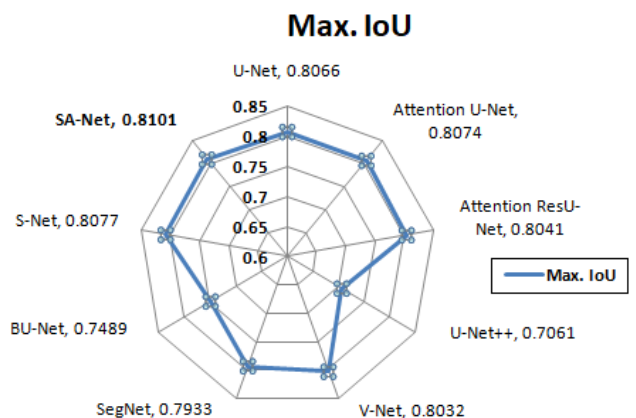


FIGURE 17. Model-wise comparison for IoU observed on LGG dataset.

IoU values of 80.77 and 81.01, respectively, despite considering the smaller model weight parameters.

- Based on our observations, we conclude that the number of model weight parameters depends on the model architecture under consideration and is directly influenced by the number of convolutional layers present in the architecture. Furthermore, introducing additional blocks such as the Attention block and/or

Residual block increases the number of model weight parameters as shown in FIGURE 22.

- The number of model weight parameters of the proposed S-Net model has decreased by 26.75%, 38.34%, 41.02%, and 30.93% from the U-Net, Attention U-Net, Attention ResU-Net, and SegNet, respectively; as the model uses less number of convolutional layers.

TABLE 5. Comparative study among different models for HGG and LGG type datasets.

Dataset Type	Model	Number of Weight Parameters (M)	Max. IoU:	Max. DSC:	Max. Sensitivity: Epoch	Max. Specificity: Epoch	Max. Accuracy: Epoch
HGG	U-Net	31.4	0.7738:100	0.7755:100	0.9593:20	0.9873:100	0.9501:100
	Attention U-Net	37.3	0.7746:100	0.7763:100	0.9415:20	0.9895:100	0.9521:100
	Attention ResU-Net	39	0.7747:100	0.7764:100	0.9481:20	0.9883:100	0.951:100
	U-Net++	9	0.6527:100	0.655:100	0.9913:50	0.7707:100	0.7423:100
	V-Net	12.6	0.7654:50	0.7675:50	0.9879:20	0.9672:75	0.9313:75
	SegNet	33.3	0.7646:100	0.7668:100	0.9906:50	0.9656:100	0.9297:100
	BU-Net	20.3	0.7321:50	0.7342:50	0.9868:75	0.9103:50	0.8764:50
	S-Net	23	0.7736:100	0.7756:100	0.9691:20	0.9834:100	0.9465:100
	SA-Net	23.3	0.7761:100	0.778:100	0.9526:20	0.9903:100	0.9529:100
LGG	U-Net	31.4	0.8066:100	0.8079:100	0.984:20	0.9871:100	0.9514:100
	Attention U-Net	37.3	0.8074:100	0.809:100	0.9672:20	0.99:100	0.9539:100
	Attention ResU-Net	39	0.8041:100	0.8059:100	0.9818:20	0.9794:100	0.9449:75
	U-Net++	9	0.7061:100	0.7078:100	0.9815:20	0.8196:100	0.7900:100
	V-Net	12.6	0.8032:100	0.805:100	0.9872:50	0.9744:100	0.9395:100
	SegNet	33.3	0.7933:100	0.7952:100	0.9837:75	0.9636:100	0.9292:100
	BU-Net	20.3	0.7489:100	0.778:20	0.9672:100	0.9467:20	0.9121:20
	S-Net	23	0.8077:100	0.8094:100	0.9599:75	0.9859:100	0.9503:100
	SA-Net	23.3	0.8101:100	0.8116:100	0.969:20	0.992:100	0.9559:100

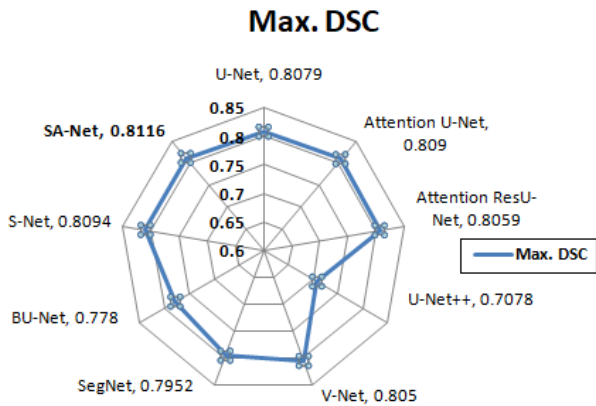


FIGURE 18. Model-wise comparison for DSC observed on LGG dataset.

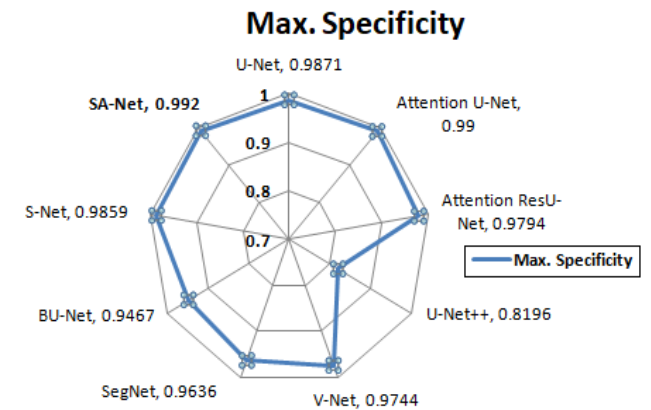


FIGURE 20. Model-wise comparison for Specificity observed on LGG dataset.

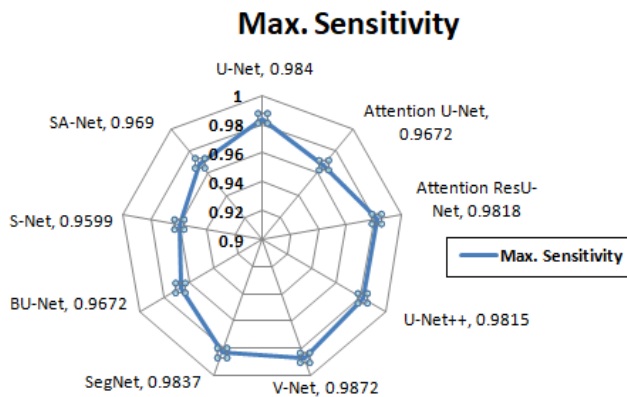


FIGURE 19. Model-wise comparison for Sensitivity observed on LGG dataset.

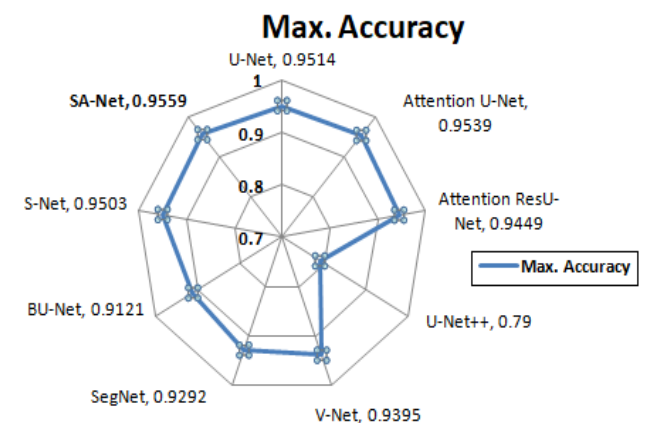


FIGURE 21. Model-wise comparison for Accuracy observed on LGG dataset.

- b) The number of model weight parameters of the proposed SA-Net model has decreased by 25.80%, 37.53%, 40.26%, and 30.03% from the U-Net, Attention U-Net, Attention ResU-Net, and SegNet models, respectively; as the model uses fewer convolutional layers.
- c) The number of model weight parameters of Attention U-Net and Attention ResU-Net have

increased by 18.79% and 24.20%, respectively, from the basic U-Net model.

- d) The number of model weight parameters of the proposed SA-Net model has increased by only 1.30% compared to the proposed S-Net model,

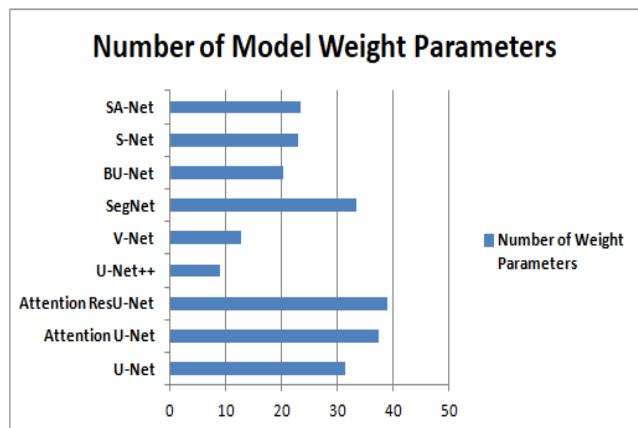


FIGURE 22. Model-wise comparison for the number of model weight parameters.

as SA-Net has utilized the concept of an ‘attention block’ to enhance the final segmentation result.

3) The proposed SA-Net model gives better performance measures (IoU, DSC, Sensitivity, Specificity, and Accuracy) for both the HGG and LGG datasets compared to all the other models we implemented in our study.

a) The performance measures for the newly proposed SA-Net model were recorded for the HGG dataset. For each performance measure, the proposed model outperformed the others under consideration as shown in FIGURES 12-16.

- The IoU measure has increased by 0.30%, 0.193%, 0.18%, 18.91%, 1.40%, 1.50%, and 6.01% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
- The DSC measure has increased by 0.32%, 0.22%, 0.21%, 18.78%, 1.37%, 1.46%, and 5.96% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
- The Sensitivity measure has increased by 1.18% and 0.47% compared to the Attention U-Net and attention ResU-Net models, respectively.
- The Specificity measure has increased by 0.30%, 0.08%, 0.20%, 28.49%, 2.39%, 2.56%, and 8.79% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
- The Accuracy measure has increased by 0.29%, 0.08%, 0.20%, 28.37%, 2.32%, 2.49%, and 8.73% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.

b) The performance measures for the newly proposed SA-Net model were recorded for the LGG

dataset below. For each performance measure, the proposed model outperformed the others under consideration as shown in FIGURES 17-21.

- The IoU measure has increased by 0.43%, 0.33%, 0.75%, 14.73%, 0.86%, 2.12%, and 8.17% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The DSC measure has increased by 0.46%, 0.32%, 0.70%, 14.66%, 0.82%, 2.06%, and 4.32% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The Sensitivity measure has increased by 1.19% compared to the Attention U-Net model.
 - The Specificity measure has increased by 0.50%, 0.20%, 1.29%, 21.03%, 1.81%, 2.95%, and 4.78% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The Accuracy measure has increased by 0.47%, 0.21%, 1.16%, 0.21%, 1.74%, 2.87%, and 4.80% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
- c) The performance measures for the newly proposed S-Net model were recorded for the HGG dataset. For each performance measure, the proposed model outperformed the others under consideration as shown in FIGURES 12-16.
- The IoU measure has increased by 18.52%, 1.07%, 1.18%, and 5.67% compared to the U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The DSC measure has increased by 0.01%, 18.41%, 1.05%, 1.15%, and 5.64% compared to the Attention U-Net, U-Net++, V-Net, SegNet, BU-Net, respectively.
 - The Sensitivity measure has increased by 1.02%, 2.93%, and 2.21% compared to the U-Net, Attention U-Net, and Attention ResU-Net models, respectively.
 - The Specificity measure has increased by 27.60%, 1.67%, 1.84%, and 8.03% compared to the U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The Accuracy measure has increased by 27.51%, 1.63%, 1.81%, and 8.00% compared to the U-Net++, V-Net, SegNet, and BU-Net, respectively.
- d) The performance measures for the newly proposed S-Net model were recorded for the LGG dataset. For each performance measure, the proposed model outperformed the others under consideration as shown in FIGURES 17-21.

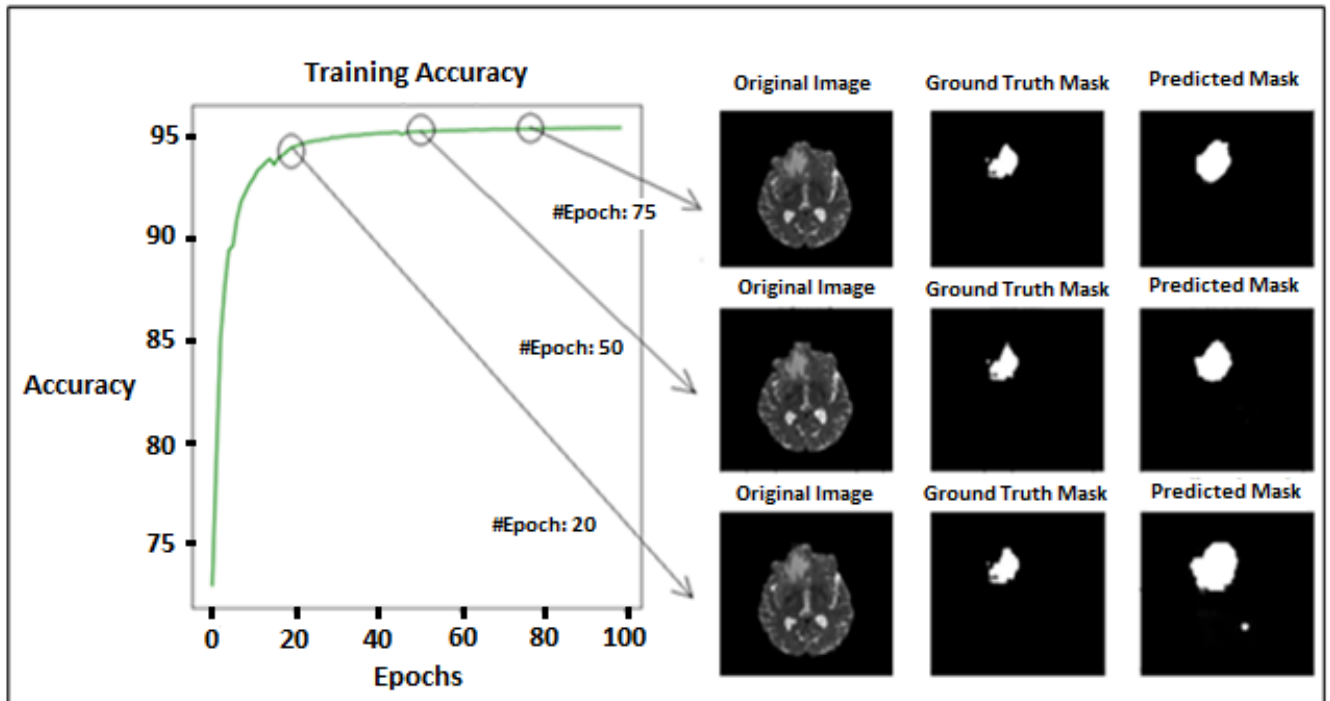
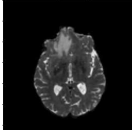
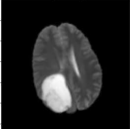


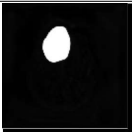
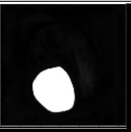
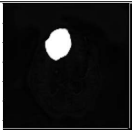
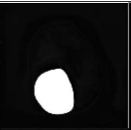
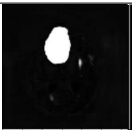


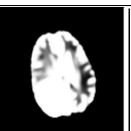






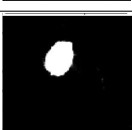

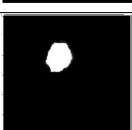
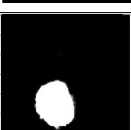


FIGURE 23. The behaviour of the proposed SA-Net model in different epochs while doing the training.

- The IoU measure has increased by 0.14%, 0.04%, 0.45%, 14.39%, 0.56%, 1.81%, and 7.85% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The DSC measure has increased by 0.18%, 0.05%, 0.43%, 14.35%, 0.55%, 1.78%, and 4.04% compared to the U-Net, Attention U-Net, Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The Specificity measure has increased by 0.66%, 20.29%, 1.18%, 2.31%, and 4.14% compared to the Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
 - The Accuracy measure has increased by 0.57%, 20.29%, 1.15%, 2.27%, and 4.19% compared to the Attention ResU-Net, U-Net++, V-Net, SegNet, and BU-Net, respectively.
- 4) The proposed models exhibited improved performance measures as the number of epochs increased during the training phase. FIGURES 24(a), 24(c), 25(a), and 25(c) clearly show that we achieved higher performance measures at higher epoch values for the HGG and LGG datasets, respectively.
- a) The proposed S-Net model has scored the following increased performance measures on the HGG dataset, as shown in FIGURE 24(a). With a gradual increase in the epoch values, we achieved better results in terms of different performance measures.
- The IoU measure has increased by 0.74%, 0.87%, and 0.95% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
 - The DSC measure has increased by 0.73%, 0.86%, and 0.94% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
 - The Specificity measure has increased by 1.05%, 1.29%, and 1.44% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
 - The Accuracy measure has increased by 1.05%, 1.27%, and 1.42% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- b) The proposed S-Net model scored the following increased performance measures observed on the LGG dataset, as shown in FIGURE 24(c). With a gradual increase in the epoch values, we achieved better results in terms of different performance measures.
- The IoU measure has increased by 1.24%, 1.37%, and 1.39% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
 - The DSC measure has increased by 1.23%, 1.35%, and 1.38% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
 - The Specificity measure has increased by 1.44%, 1.62%, and 1.68% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.

TABLE 6. Sampled images and their corresponding ground truth and predicted output segmentation masks generated by each model.

Model	HGG Dataset	LGG Dataset
Image		
Ground Truth		
U-Net		
Attention U-Net		
Attention Res U-Net		
U-Net++		
V-Net		
SegNet		
BU-Net		
S-Net		
SA-Net		

- The Accuracy measure has increased by 1.44%, 1.61%, and 1.68% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.

c) As shown in FIGURE 25(a), the performance measures for the newly proposed SA-Net model are recorded for the HGG dataset below. The proposed model outperformed the other models considered for each performance measure.

- The IoU measure has increased by 0.44%, 0.50%, and 0.53% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- The DSC measure has increased by 0.43%, 0.52%, and 0.53% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- The Specificity measure has increased by 0.65%, 0.88%, and 0.91% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- The Accuracy measure has increased by 0.63%, 0.86%, and 0.88% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.

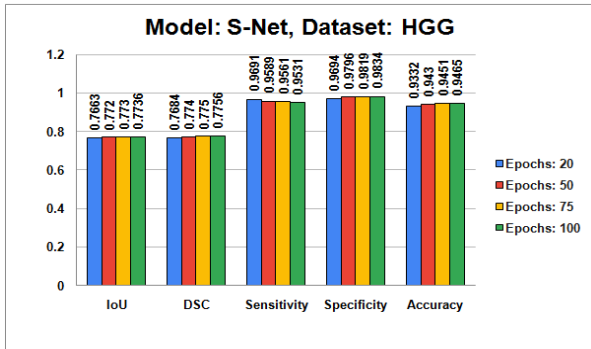
d) As shown in FIGURE 25(c), the performance measures for the newly proposed SA-Net model are recorded for the LGG dataset. The proposed model outperformed the other models and considered the following performance measures.

- The IoU measure has increased by 1.33%, 1.41%, and 1.44% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- The DSC measure has increased by 1.30%, 1.37%, and 1.40% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- The Specificity measure has increased by 1.87%, 2.07%, and 2.15% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.
- The Accuracy measure has increased by 1.85%, 2.04%, and 2.11% at epochs 50, 75, and 100, respectively, w.r.t. epoch 20.

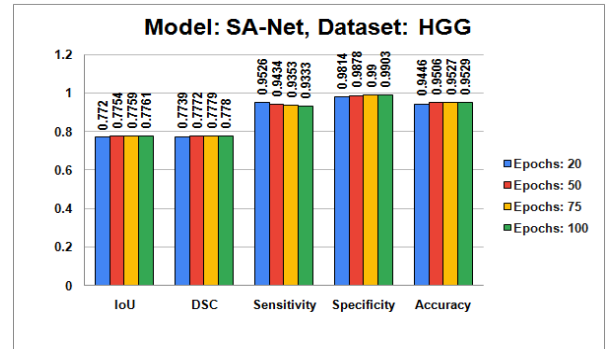
5) The proposed S-Net and SA-Net models give better performance measures for the dropout rate of 0.2 applied only to the middle layers than the dropout rate of 0.5 applied on the middle layers and the mixed dropout rate of 0.2 at the input layer and 0.5 on both the hidden and output layers, respectively as shown in FIGURES 24(b), 24(d), 25(b), and 25(d). In the following text, the mixed mode of 0.2 and 0.5 is represented as (0.2 + 0.5). Furthermore, while tuning the dropout rate, we considered the model weights only for the epoch values of 20.

a) As shown in FIGURE 24(b), the proposed S-Net model scored the following improved performance measures observed on the HGG dataset.

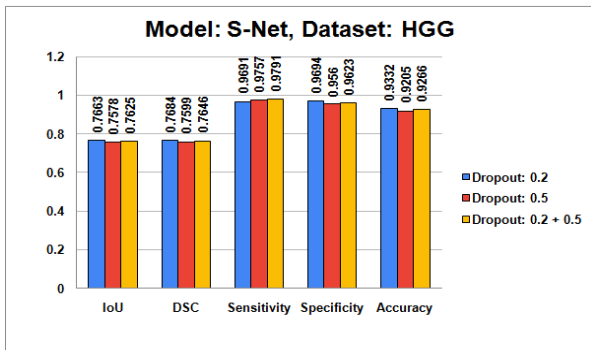
- The IoU measure has increased with a dropout rate of 0.2 by 1.12% and 0.50% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
- The DSC measure has increased with a dropout rate of 0.2 by 1.12% and 0.50% from the



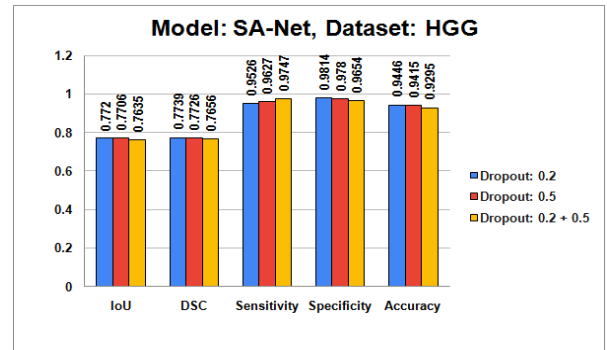
(a) Epoch-wise performance measures for the S-Net model observed on the HGG dataset.



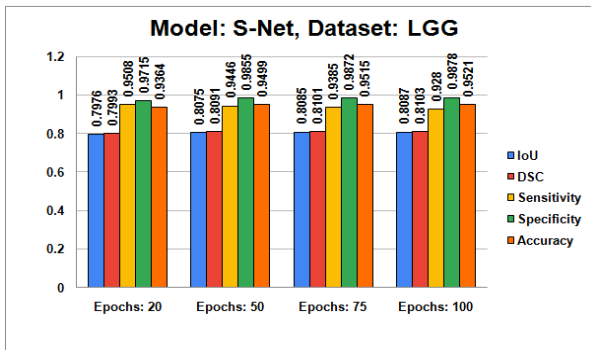
(a) Epoch-wise performance measures for the SA-Net model observed on the HGG dataset.



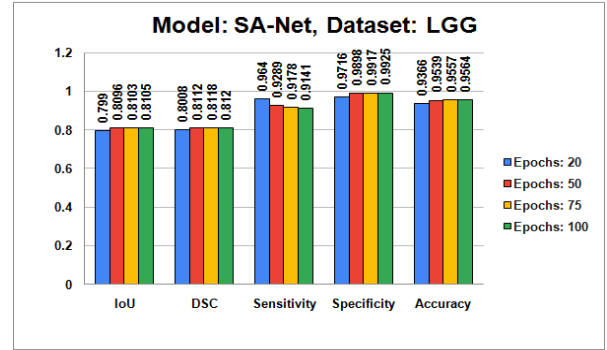
(b) Dropout-wise performance measures for the S-Net model observed on the HGG dataset.



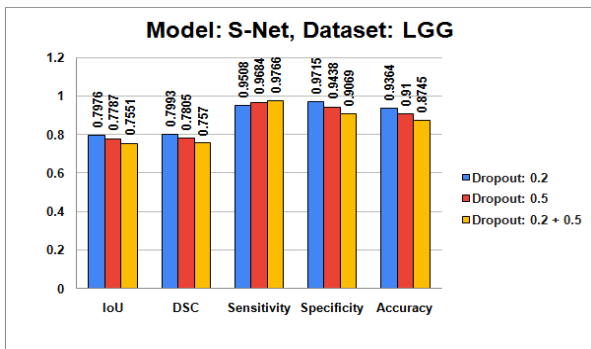
(b) Dropout-wise performance measures for the SA-Net model observed on the HGG dataset.



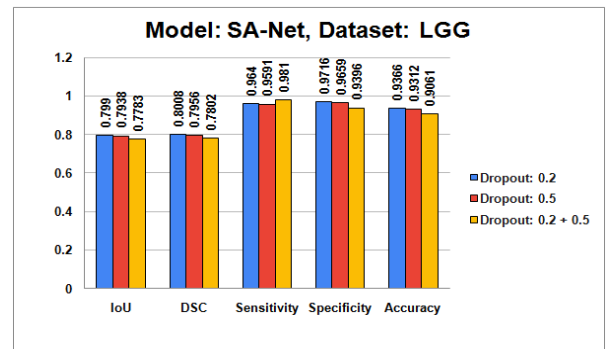
(c) Epoch-wise performance measures for the S-Net model observed on the LGG dataset.



(c) Epoch-wise performance measures for the SA-Net model observed on the LGG dataset.



(d) Dropout-wise performance measures for the S-Net model observed on the LGG dataset.



(d) Dropout-wise performance measures for the SA-Net model observed on the LGG dataset.

FIGURE 24. Different observations for the S-Net model.

FIGURE 25. Different observations for the SA-Net model.

dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.

- The Specificity measure has increased with a dropout rate of 0.2 by 1.40% and 0.74% from

- the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
- The Accuracy measure has increased with a dropout rate of 0.2 by 1.38% and 0.71% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
- b) As shown in FIGURE 24(d), the proposed S-Net model scored the following enhanced performance measures observed on the LGG dataset.
- The IoU measure has increased with a dropout rate of 0.2 by 2.43% and 5.63% from a dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The DSC measure has increased with a dropout rate of 0.2 by 2.41% and 5.59% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The Specificity measure has increased with a dropout rate of 0.2 by 2.93% and 7.12% from a dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The Accuracy measure has increased with a dropout rate of 0.2 by 2.90% and 7.08% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
- c) As shown in FIGURE 25(b), the proposed SA-Net model scored the following enriched performance measures observed on the HGG dataset.
- The IoU measure has increased with a dropout rate of 0.2 by 0.18% and 1.11% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The DSC measure has increased with a dropout rate of 0.2 by 0.17% and 1.08% from a dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The Specificity measure has increased with a dropout rate of 0.2 by 0.35% and 1.66% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The Accuracy measure has increased with a dropout rate of 0.2 by 0.33% and 1.62% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
- d) As shown in FIGURE 25(d), the proposed SA-Net model scored the following upgraded performance measures observed on the LGG dataset.
- The IoU measure has increased with a dropout rate of 0.2 by 0.65% and 2.66% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
 - The DSC measure has increased with a dropout rate of 0.2 by 0.65% and 2.64% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.

- The Specificity measure has increased with a dropout rate of 0.2 by 0.59% and 3.40% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.
- The Accuracy measure has increased with a dropout rate of 0.2 by 0.58% and 3.37% from the dropout rate of 0.5 and a mixed dropout rate of (0.2 + 0.5), respectively.

V. CONCLUSION AND FUTURE SCOPE

In this paper, we have introduced two new models, S-Net and SA-Net, to perform the brain tumour segmentation task in a binary mode. Both the proposed S-Net and SA-Net model architectures have used U-Net as the baseline architecture. The primary reason for selecting U-Net-based models for comparison is because historically U-Net-based models have provided better results in the field of medical image segmentation tasks. The concepts of ‘Merge block’ and ‘Attention block’ are also used by these proposed model architectures to get higher performance measures. Moreover, the concept of a ‘Merge block’ is used to concatenate all the features from all the preceding layers both in the ‘contracting path’ and ‘expansive path’ by using a limited number of training samples populated utilizing various data augmentation techniques. The proposed SA-Net model has incorporated the concept of the ‘Attention block’ after the ‘Merge block’ so that the performance gets enhanced by focusing on the area of interest having a tumour region. The proposed models are evaluated on BraTS 2018, BraTS 2019, BraTS 2020, and BraTS 2021 datasets. Both models have exhibited good improvement considering various performance measures (IoU, DSC, Sensitivity, Specificity, and Accuracy) when compared with the baseline U-Net architecture. In our work, we faced difficulties due to the limited available computing environment; hence, the experiment was limited to a maximum of 100 epochs only. In the future, we intend to explore further by increasing the number of epochs as part of an extension of our proposed work. Additionally, we intend to explore the 3D-based networks to improve the performance of the proposed segmentation models. Furthermore, the process of hyperparameter tuning can also be enhanced by using grid search and random search-based techniques. In the future, we have plans to extend our experiment by incorporating the comparison of newly proposed models with non-U-Net-based models.

REFERENCES

- [1] D. Zikic, Y. Ioannou, M. Brown, and A. Criminisi, “Segmentation of brain tumor tissues with convolutional neural networks,” in *Proc. MICCAI-BRATS*, vol. 36, 2014, pp. 36–39.
- [2] G. Urban, M. Bendszus, F. Hamprecht, and J. Kleesiek, “Multi-modal brain tumor segmentation using deep convolutional neural networks,” in *Proc. Multimodal Brain Tumor Segmentation (BraTS) Challenge of Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Boston, MA, USA: Harvard Medical School, 2014, pp. 31–35.
- [3] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, “Deep convolutional neural networks for the segmentation of gliomas in multi-sequence MRI,” in *Proc. BrainLes*. Cham, Switzerland: Springer, 2015, pp. 131–143.
- [4] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, 2015, pp. 234–241.

- [5] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1240–1251, May 2016.
- [6] T. K. Lun and W. Hsu, "Brain tumor segmentation using deep convolutional neural network," in *Multimodal Brain Tumor Segmentation (BraTS) Challenge of Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Munich, Germany, 2016.
- [7] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.* Cham, Switzerland: Springer, 2016, pp. 424–432.
- [8] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.
- [9] V. Alex, M. Safwan, and G. Krishnamurthi, "Brain tumor segmentation from multi modal MR images using fully convolutional neural network," in *Medical Image Computing and Computer Assisted Intervention—MICCAI*, 2017, pp. 1–8.
- [10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [11] E. Caver, L. Chang, W. Zong, Z. Dai, and N. Wen, "Automatic brain tumor segmentation using a U-Net neural network," in *Proc. Pre-Conf. 7th MICCAI BraTS Challenge*, vol. 63, 2018, pp. 63–73.
- [12] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.
- [13] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation," 2018, *arXiv:1802.06955*.
- [14] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 3–11.
- [15] Y. Zhou, W. Huang, P. Dong, Y. Xia, and S. Wang, "D-UNet: A dimension-fusion U shape network for chronic stroke lesion segmentation," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 3, pp. 940–950, May 2021.
- [16] C. Wu, Y. Zou, and Z. Yang, "U-GAN: Generative adversarial networks with U-Net for retinal vessel segmentation," in *Proc. 14th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2019, pp. 642–646.
- [17] S. Sajid, S. Hussain, and A. Sarwar, "Brain tumor detection and segmentation in MR images using deep learning," *Arabian J. Sci. Eng.*, vol. 44, no. 11, pp. 9249–9261, Nov. 2019.
- [18] X. Feng, C. Wang, S. Cheng, and L. Guo, "Automatic liver and tumor segmentation of CT based on cascaded U-Net," in *Proc. Chin. Intell. Syst. Conf.* Singapore: Springer, 2019, pp. 155–164.
- [19] W. Yu, B. Fang, Y. Liu, M. Gao, S. Zheng, and Y. Wang, "Liver vessels segmentation based on 3D residual U-NET," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 250–254.
- [20] M. Kolarik, R. Burget, V. Uher, and L. Povoda, "Superresolution of MRI brain images using unbalanced 3D dense-U-Net network," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2019, pp. 643–646.
- [21] Z. Zhang, C. Wu, S. Coleman, and D. Kerr, "DENSE-inception U-Net for medical image segmentation," *Comput. Methods Programs Biomed.*, vol. 192, Aug. 2020, Art. no. 105395.
- [22] W. H. Khoong, "BUSU-Net: An ensemble U-Net framework for medical image segmentation," 2020, *arXiv:2003.01581*.
- [23] M. U. Rehman, S. Cho, J. H. Kim, and K. T. Chong, "BU-Net: Brain tumor segmentation using modified U-Net architecture," *Electronics*, vol. 9, no. 12, p. 2203, Dec. 2020.
- [24] O. Petit, N. Thome, C. Rambour, L. Themyr, T. Collins, and L. Soler, "U-Net transformer: Self and cross attention for medical image segmentation," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Cham, Switzerland: Springer, 2021, pp. 267–276.
- [25] X. Liu, L. Song, S. Liu, and Y. Zhang, "A review of deep-learning-based medical image segmentation methods," *Sustainability*, vol. 13, no. 3, p. 1224, Jan. 2021.
- [26] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.
- [27] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, "A review on deep learning in medical image analysis," *Int. J. Multimedia Inf. Retr.*, vol. 11, no. 1, pp. 19–38, 2022.
- [28] Spyridon (Spyros) Bakas, "Brats MICCAI brain tumor dataset," Center Biomed. Image Comput. Anal. (CBICA), SBIA, UPenn, PA, USA.
- [29] H. M. Luu and S.-H. Park, "Extending nn-UNet for brain tumor segmentation," 2021, *arXiv:2112.04653*.
- [30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [31] B. Hakyemez, C. Erdogan, I. Ercan, N. Ergin, S. Uysal, and S. Atahan, "High-grade and low-grade gliomas: Differentiation by using perfusion MR imaging," *Clin. Radiol.*, vol. 60, no. 4, pp. 493–502, Apr. 2005.
- [32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [33] A. Labach, H. Salehinejad, and S. Valaee, "Survey of dropout methods for deep neural networks," 2019, *arXiv:1904.13310*.
- [34] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-UNet: UNet-like pure transformer for medical image segmentation," 2021, *arXiv:2105.05537*.
- [35] M. Futrega, A. Milesi, M. Marcinkiewicz, and P. Ribalta, "Optimized U-Net for brain tumor segmentation," 2021, *arXiv:2110.03352*.
- [36] S. S. Al-Amri, N. V. Kalyankar, and S. D. Khamitkar, "Image segmentation by using threshold techniques," 2010, *arXiv:1005.4020*.
- [37] S. Zhu, X. Xia, Q. Zhang, and K. Belloulata, "An image segmentation algorithm in image processing based on threshold segmentation," in *Proc. 3rd Int. IEEE Conf. Signal-Image Technol. Internet-Based Syst.*, Dec. 2007, pp. 673–678.
- [38] F. Jiang, M. R. Frater, and M. Pickering, "Threshold-based image segmentation through an improved particle swarm optimisation," in *Proc. Int. Conf. Digit. Image Comput. Techn. Appl. (DICTA)*, Dec. 2012, pp. 1–5.
- [39] V. W. Stieber, "Low-grade gliomas," *Current Treatment Options Oncol.*, vol. 2, no. 6, pp. 495–506, 2001.
- [40] A. Fabijańska, "Variance filter for edge detection and edge-based image segmentation," in *Proc. Perspective Technol. Methods MEMS Design*, 2011, pp. 151–154.
- [41] M. J. Islam, S. Basalamah, M. Ahmadi, and M. A. Sid-Ahmed, "Capsule image segmentation in pharmaceutical applications using edge-based techniques," in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, May 2011, pp. 1–5.
- [42] M. M. S. J. Preetha, L. P. Suresh, and M. J. Bosco, "Image segmentation using seeded region growing," in *Proc. Int. Conf. Comput., Electron. Electr. Technol. (ICCEET)*, Mar. 2012, pp. 576–583.
- [43] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.
- [44] B. J. Theeler and M. D. Groves, "High-grade gliomas," *Current Treatment Options Neurol.*, vol. 13, no. 4, pp. 386–399, 2011.
- [45] N. Dhanachandra, K. Mangleam, and Y. J. Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm," *Proc. Comput. Sci.*, vol. 54, pp. 764–771, Jan. 2015.



SUNITA ROY received the B.Sc. degree in computer science from Barrackpore Rastraguru Surendranath College, in 2006, and the M.Sc. degree in computer and information science and the M.Tech. degree in computer science and engineering from the University of Calcutta, India, in 2008 and 2010, respectively. She has been a Faculty Member with the Department of Computer Science, Lady Brabourne College, Kolkata, India, since 2010. She has been doing part-time research

work as a registered Research Scholar with the Department of Computer Science and Engineering, University of Calcutta, since 2011. She has authored more than 12 research articles, published in different journals and conference proceedings. Her research interests include machine learning and computer vision applied in the biomedical engineering domain like brain tumor detection. She is also investigating techniques related to face detection and its recognition.



RIKAN SAHA received the M.Sc. degree in computer science and the M.Tech. degree in computer science and engineering from the University of Calcutta, India, in 2020 and 2022, respectively. His research interests include wireless sensor networks, data science, machine learning, and computer vision.



SUVARTHI SARKAR received the M.Sc. degree in computer and information science and the M.Tech. degree in computer science and engineering from the University of Calcutta, India, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with IIT Guwahati, India. His research interests include cloud computing, machine learning, financial analysis, and computer vision.



RANJAN MEHERA received the B.Sc. degree in computer science, the M.Sc. degree in computer and information science, the M.Tech. degree in computer science and engineering, and the Ph.D. degree in computer science and engineering from the University of Calcutta, India, in 2003, 2005, 2007, and 2016, respectively. He is currently a Solutions Engineer with Anodot Inc., where he leads the Business Development Team for the Anodot products (Anomaly Detection and Cloud Cost Management) focused in the American region. He is a trusted advisor for all levels of the organization from engineers to CxOs, bringing over a decade of experience in software engineering, network architectures, and product development. He has authored more than 15 technical research articles. He also holds several international patents. His part-time research interests include computer science, machine learning, business analytics, graph theory, computational geometry, and VLSI designs.



RAJAT KUMAR PAL (Member, IEEE) received the B.E. degree in electrical engineering from the Bengal Engineering College, Shibpur, University of Calcutta, India, in 1985, the M.Tech. degree in computer science and engineering from the University of Calcutta, in 1988, and the Ph.D. degree in VLSI physical design from IIT Kharagpur, India, in 1996. He has been a Faculty Member with the Department of Computer Science and Engineering, University of Calcutta, since 1994. He also served as a Professor with the Information Technology Department, Assam University, from 2010 to 2012. Currently, he is a Professor with the Department of Computer Science and Engineering, University of Calcutta. He has published more than 250 research articles and has authored two books. He also holds several international patents. His major research interests include VLSI designs, graph theory and its applications, perfect graphs, logic synthesis, design and analysis of algorithms, computational geometry, and parallel computation and algorithms.



SAMIR KUMAR BANDYOPADHYAY (Senior Member, IEEE) received the B.E. degree in electronics and telecommunication engineering specialization from the Bengal Engineering College, Shibpur, University of Calcutta, India, the M.Tech. degree in radio physics and electronics and the Ph.D. degree in computer science and engineering from the University of Calcutta, India, in 1979 and 1988, respectively, and the M.B.B.S. degree from the Calcutta Medical College, University of Calcutta, in 1989. He is a retired Professor with the Department of Computer Science and Engineering, University of Calcutta, India. He has authored books titled *Data Structure Using C* (Addison Wesley, 2003) and *C Language* (Pearson Publication, 2010). He also authored more than hundreds of research articles in national and international journals and conference proceedings. He acted as the Chairperson of many forums, such as the Science and Engineering Research Support Society (SERSC, Indian Part), a fellow of the Computer Society of India, and the Sectional President of ICT of the Indian Science Congress Association, from 2008 to 2009. He is a member of ACM and a fellow of the Institution of Engineers, India, and the Institution of Electronics and Telecommunication Engineering, India.

...