Received: November 29, 2021. Revised: December 24, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

SDA-UNET2.5D: Shallow Dilated with Attention Unet2.5D for Brain Tumor Segmentation

Agus Subhan Akbar[1,2] Chastine Fatichah[1]* Nanik Suciati[1]

[1]*Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
[2]*Department of Information System, Faculty of Science and Technology, Universitas Islam Nahdlatul Ulama Jepara, Jepara, Indonesia*
* Corresponding author's Email: chastine@if.its.ac.id

Abstract: Many studies have been carried out to segmentation brain tumors on 3D Magnetic Resonance Imaging (MRI) images with 3D or 2D approaches. The 3D approach pays attention to the interrelationships between slices in a 3D image. However, this requires high resources, while the 2D approach requires lower resources but ignores the voxel relationship in 3D space. The 2.5D approach seeks to combine the lightness of the 2D approach and the voxel interconnection of the 3D approach. This article proposes SDA-UNet2.5D, a shallow UNet 2.5D architecture that pays attention to the interconnectedness of 3D images by involving five slices to get one slice of segmentation prediction results. The architecture is trained using the Brain Tumor Segmentation (BraTS) 2018, 2019, and 2020 datasets. Compared to other architectures, this proposed architecture has a high segmentation speed with 4.05-4.24 seconds to segment one patient data. Online validation resulted in superior average dice performance of 75.70, 88.82, 77.33 for the BraTS 2018, 71.29, 88.00, 76.55 for the BraTS 2019, and 70.80, 87.95, 75.89 for the BraTS 2020 validation datasets in the areas of Enhanced Tumor (ET), Whole Tumor (WT), and Tumor Core (TC).

Keywords: Shallow unet 2.5d, Atrous convolution, Attention mechanisms, Brain tumor segmentation, 2.5D approach.

1. Introduction

Automatic brain tumor segmentation is one way to obtain information on the tumor part in the brain for further medical action by utilizing computer computing. This method is taken to avoid manual segmentation, which needs experts, is time-consuming, and is error-prone. Automatic segmentation is faster, produces consistent results compared to when an expert processes it at a different time or with another person.

One of the brain tumor datasets that have been labelled for deep learning-based learning is the Brain Tumor Segmentation 2018 Challenge datasets [1, 2]. The dataset was provided along with the challenge that was held in 2018. The dataset consists of the training data and the validation data. Each dataset contains four modalities of 3D image data from MRI scans with different retrieval protocols (T1, T2, T1ce, and Flair) with a size of $240 \times 240 \times 155$. The training data is equipped with ground truth for training purposes, while the validation data is not equipped with ground truth. The validation data segmentation results can only be validated using the online validation tool at https://ipp.cbica.upenn.edu. The BraTS dataset is growing rapidly with challenges in the following years. The latest available datasets besides BraTS 2018 are BraTS 2019, BraTS 2020, and BraTS 2021, with an increasing number of data [3-5].

The automatic segmentation process for 3D images has been carried out in many studies with some different approaches. The 2D processing approach is used in [6] by breaking the 3D image into several slices and normalizing the pixel values in the interval 0-255, manipulating the least significant bit (LSB) and the most significant bit (MSB) of the image. This processing approach is fast because it

processes 2D images but ignores the interrelationships between slices of 3D images so that the performance is not optimal. Another approach is taken by Hu et al. [7] which processes 2D slices of each of the three axes: axial, coronal, and sagittal and recombine each of the outputs into one. This approach yields good performance, but the training will be performed three times, and the inference process for each slice on each axis also be performed three times. Rezaei et al. [8] use 3D approach by using two 3D UNet arranged sequentially. Because this 3D approach requires a large memory capacity, the 3D image is broken into $64 \times 64 \times 64$ size.

This paper proposes an architecture that focuses on segmenting brain tumors from 3D images with a 2.5D approach while keeping the interrelationships between slices in 3D images and small memory requirements. By processing some slices of the four existing modalities via a 3d to 2d converter block, 3D images can be processed with some 2D convolution blocks that take advantage of atrous convolution and attention gate, resulting in better segmentation performance. This 2D convolution block makes the size of the processed image pieces can be more expansive, expected to improve the segmentation performance. In summary, the contributions in this paper are given as follows:

1. We designed the Shallow dilated with attention UNet2.5D architecture with block modifications that utilize atrous convolutions and attention mechanism to do brain tumor segmentation
2. We proposed a Multi Dilated Residual with Attention Mechanism block that replace block processing in transfer section in the UNet architecture. Dilated factors used in the block contains sequence of 1,2,4, and 8.
3. We proposed d-dilated residual block, a residual block with *d* factor/parameter of dilated used in atrous convolution.

Furthermore, this paper is organized into the following sections: Related Study is described in Section 2. Section 3 contains the material and proposed method used in this study. Section 4 presents the results achieved with their analysis and a comparison of performance with the four current methods. Section 5 contains conclusions and future works.

2. Related works

Deep Convolutional Neural Networks (DCNNs) have been widely used in medical image segmentation. UNet [9] is one of the deep learning architectures used to perform this automatic segmentation. This architecture consists of 3 main parts: the contracting section, the transfer section, and the expanding section. What distinguishes this architecture from other FCN architectures is a skip connection section that connects the contracting section with the corresponding expanding section. Subsequent researchers developed this UNet architecture with several modifications. UNet development with replacement of block contents is done at [10-13]. The replacements include residual block [11, 13, 14], two-path residual blocks and applying variations in the use of these blocks in the contracting and expanding sections [10], and dilated convolution [12, 15]. In addition, the skip connection section has also been modified with an attention gate at [13].

A residual block is a way to connect the first part of a block with the last part of the block. This method is used to avoid vanishing gradients in the deeper blocks of the deep convolutional neural network. Therefore the architecture can be structured more deeply without losing the gradient [16]. The placement of the activation function in the residual block that produces the best performance is strategic pre-activation compared to other placement variations [17].

One of the essential factors in the convolution process is the size of the receptive field used. The receptive field is the size of the area in the input section that is used to form the output feature [18]. The larger the size of the receptive field used, the more information that can be used to form the output feature, and the smaller the input information that is not used to form the output feature [19]. However, the larger the receptive field, the larger the number of DCNN architecture parameters. One way to increase the size of the receptive field without increasing the number of parameters is to use atrous convolution. The use of atrous convolution has succeeded in increasing performance as reported in [12, 15, 20].

Small objects in the image to be segmented usually fail to be segmented. One method to minimize the possibility of missing segmentation is to apply attention. Attention is a way of focusing processing on only the important parts during training. Thus, it can reduce processing on the parts that are not relevant. Attention was used in [21-23] and reported good results.

Regarding detecting brain tumors, many studies have been carried out to detect their presence. Tjahyaningtijas et al. [24] proposed the en-CNN architecture, which is a modification of VGG16 to detect the presence of tumors using the BraTS 2018 dataset. This architecture uses a 2D approach by taking slices of the available modalities and reducing their size into $224 \times 224 \times 1$ to be processed by the

en-CNN architecture. The output of the en-CNN architecture is a binary class in the form of high-grade glioma and low-grade glioma. Furthermore, Rao et al. [25] proposed an approach to segmentation of the presence of tumors as a whole before classifying tumors using the Hybrid Kernel-based Fuzzy C-Means-Convolutional Neural Network (KFCM-CNN) method using the T1-Weighted Contrast Enhanced MRI dataset. The entire tumor area was first segmented by KFCM followed by feature extraction to obtain input features for CNN. The final output of this method is in the form of three tumor classes consisting of meningioma, glioma, and pituitary. These two approaches focus more on tumor classification. The first approach does not segment the tumor area first, while the second approach segments the entire tumor area as an input to the classification architecture using CNN.

In processing brain tumor segmentation with datasets from BraTS, Benson et al. [26] use a 2D approach by stacking several convolution layers and utilizing bit processing for 2D image management. Tuan et al. [6] also adopt a 2D UNet architecture approach by implementing two kernel sizes, 3×3 and 5×5. Furthermore, Lorenzo et al. [14] replaced the plain convolution in 2D UNet with residual blocks and used two consecutive UNet models to process tumor segmentation. The first UNet was used to segment the entire tumor area, and the second UNet was used to segment it into three predefined classes. Kotowski et al. [27] uses the same approach by detecting all tumor areas followed by multiclass classification of pixels that have been detected as tumors. This 2D processing approach for 3D images has the advantage of using minimal GPU memory footprint but eliminating voxel connectivity information in 3D space.

The adoption of 3D processing was carried out by Bhalerao et al. [28] by modifying the UNet 2D architecture from [9] into 3D architecture and replacing its processing block to residual block with two convolutions. Another approach is taken by Ahmad et al. [29] by replacing UNet3D blocks with residual dense blocks to minimize the number of parameters and add atrous-spatial pyramid pooling blocks at each level of the expanding part of UNet to capture a multilevel contextual feature map. Kotowski et al. [30] apply a 3D approach by cascading two UNet architectures, one for the detection of whole tumors and the other for performing multiclass classification according to the components of tumor formation. Rezaei et al. [31] modified UNet 3D and used it as a generator in the GAN architecture used for tumor segmentation. Furthermore, Rezaei et al. [8] used two 3D UNet architectures as generators in developing the GAN architecture for tumor segmentation. Agravat et al. [32] developed the FCNN 3D encoder-decoder architecture to perform tumor segmentation. The application of UNet 3D with several levels of downsampling has given better results. However, several levels of downsampling have reduced the precision of spatial/position information from the tumor. In addition, the use of the 3D approach also requires greater resources than the 2D approach.

The 2.5D processing approach is intended to overcome the absence of voxel interconnection in the 2D approach while minimizing the need for high processing devices in the 3D approach. Zhao et al. [33] replaces its UNet3D processing block by applying three 2D convolutions to replace one 3D convolution. Each 2D convolution is used to process the map features of each of the axial, sagittal, and coronal axes. Hu et al. [7] uses a 2.5D approach by processing 3D medical images from each of the axial, coronal, and sagittal axes, improving the quality of the segmentation with CRF and combining the results from each axis to get the final segmentation. Indraswari et al. [34] also uses a three-axis projection approach to segment the entire tumor area without dividing it in detail into each tumor component. The approach of combining each of these axes results in good segmentation performance but requires training and inference as many as three axes are used. Image processing is also performed three times from each axis before being combined into one final segmentation result.

The selection of the final segmentation layer kind, the image crop size used, and the post-processing segmentation layer significantly affects architectural inference speed. A fully connected layer in the final segmentation layer increases the number of architectural parameters trained compared to using a convolution layer. A small image crop size will require more processing iterations than a more significant one. Furthermore, an additional complex post-processing layer will increase the time required for single data inference. Pereira et al. [35] proposed a CNN architecture that uses a 2D approach with patch-based processing of 33×33 to produce a one-pixel segmentation output with five probability values for each class (normal tissue, necrosis, edema, non-enhancing, and tumor enhancing). The final three layers of the proposed architecture are fully connected layers with the last layer output using the softmax activation function. On the other hand, Havaei et al. [36] proposes the InputCascadeCNN architecture, which combines two TwoPathCNNs with patch sizes of 33×33 and 65×65. TwoPathCNN contains two processing paths a local

path (with a 7 × 7 kernel size) and a global path (with a 13 × 13 kernel size). The final output is processed using a convolution layer that produces five class probability values for each segmented voxel target. Hu et al. [7] developed InputCascadeCNN and applied it to three axes, axial, sagittal, and coronal, improving the segmentation results with Conditional Random Fields (CRF) and combining the outputs of each axis with a voting strategy. Furthermore, Zhao et al. [37] uses a 2.5D approach using a Fully Convolutional Neural Network (FCNN) architecture connected to a CRF, formulated as a Recurrent Neural Network (CRF-RNN). The FCNN developed is similar to [36] but only uses three of the four MRI image modalities and processes from three axes and combines the final output using a voting strategy. Chen et al. [38] developed the DeepMedic architecture by combining features maps from several previous convolution layers to enrich the features that have been studied to be incorporated into the multi-layer perceptron architecture as a post-processing method to produce the final segmentation output. This architecture uses a 3D approach with a crop size of $25 \times 25 \times 25$. The use of fully connected layers in [35], the small crop size in [7, 35-38], and the additional use of complex post-processing such as in [37, 38] cause each architecture to require significant inference time for processing a single patient data.

Increasing the receptive field area using atrous convolution and attention mechanism in tumor segmentation has given promising results. Awasthi et al. [39] apply attention gate to skip connection in UNet 2D and perform segmentation for each target using one independent model. In addition to implementing an attention gate in the skip connection section, Xu et al. [40] also adds supervision in the segmentation layer in the expanding layer. Savadikar et al. [41] used a 2D probabilistic UNet and the application of an attention mechanism on the skip connection section. Guo et al. [42] applied a combination of two patch sizes and two types of attention (spatial attention and attention channel) and mixed them on UNet 3D to form four independent models for segmenting tumors. Meanwhile, Yan et al. [43] implements a Squeeze-and-Excitation block (SEB) in each residual block in the 3D encoder-decoder architecture, which is used to improve segmentation performance. The SEB is also used by Zhao et al. [33] to improve the segmentation performance of their model. The use of atrous convolution and attention mechanisms separately or together has improved the performance of tumor segmentation models. The combination of using atrous convolution and attention mechanism in one architecture is expected to improve model segmentation performance further.

This article proposes the Shallow Dilated with Attention Unet2.5D architecture to overcome the previous problems. This architecture is designed with a 2.5D approach to maintaining the relationship between voxels in 3D space, consists of one level of downsampling, utilizes the power of atrous convolution and attention mechanisms, and uses simple but powerful post-processing. The 2.5D approach is made by taking five slices of a 3D image and converting them into a 2D feature map that can be processed with 2D convolution, which is faster, lighter, and requires less GPU RAM compared to 3D convolution. The 3D to 2D converter block used differs from the 2.5D processing of other architectures, which incorporate the processing of slices from three axes. Utilization of one level of downsampling is intended to minimize the shift in spatial/position information from the tumor due to the pooling process. Atrous convolution is intended to expand the receptive area without using a large kernel size. Furthermore, the use of the attention mechanism is intended to strengthen features related to segmentation targets, especially for small objects. Combining the atrous convolution arrangement and attention mechanism in a block significantly affects the segmentation performance. Finally, simple post-processing is intended to simplify the conversion process of the final processing results to the reconstruction of the segmentation target.

3. Material and method

3.1 Dataset

The datasets used in this study are BraTS 2018, BraTS 2019, and BraTS 2020 datasets. Each dataset consists of a training and validation dataset. The BraTS 2018 training dataset contains 285 patient data, with four modalities (T1, T2, T1ce, Flair, and one label). In comparison, the BraTS 2018 validation dataset contains 66 patient data with four modalities without any ground truth. The BraTS 2019 training dataset contains 335 patient data with four modalities and labels, while the corresponding validation dataset contains 125 patients with four modalities without segmentation labels. Furthermore, The BraTS 2020 training dataset consisted of 369 patient data with four modalities and segmentation labels, while the validation dataset contained 125 data without segmentation labels.

For local training and validation purposes, the training dataset is divided into two parts with a composition of 80:20—80% for training, and 20%

Table 1. Maximum volume of BraTS 2018 image cropped

Dataset	Maximum volume size
Training Data	$154 \times 187 \times 149$
Validation Data	$164 \times 190 \times 147$

Table 2. Range of voxel intensity values of BraTS 2018 dataset

Dataset	Modality	Minimum Value	Maximum Value
Training Data	T1	0.0	32767.0
	T2	0.0	32767.0
	T1ce	0.0	32767.0
	Flair	0.0	32767.0
Validation Data	T1	0.0	32767.0
	T2	0.0	32767.0
	T1ce	0.0	32767.0
	Flair	-36.1	32767.0

for local validation data. The 5-Fold Cross Validation strategy was used to stabilize the training results.

For online validation purposes, the validation dataset is processed using the trained model. The processing results are stored in medical file format (.nii.gz) to follow the prerequisites determined by the online validation tool.

3.2 Pre-processing

The pre-processing carried out on the data consists of two things. The first is cropping the image in order to get a volume that contains only the brain. From direct checking on the BraTS 2018 dataset, the maximum volume obtained is listed in Table 1.

This cropping results in reduced memory requirements during training and inference models. Starting location of the cropping is still carried out to reconstruct the original size ($240 \times 240 \times 155$) for online validation.

The voxel value of the MRI image does not follow the general standard for images with gray values in the range 0-255. From directly checking the voxel values for each modality in the BraTS 2018 dataset listed in Table 2.

By taking into account these varying values, these values are normalized first using Eq. (1)

$$C_{norm} = \frac{C_{orig} - \mu}{\sigma + \epsilon} \quad (1)$$

With C_{orig} and C_{norm} representing the original and the normalized image, μ and σ are the mean and standard deviation values of the voxels in the normalized area, and ϵ is the small numbers to prevent division by zero. This normalization applied to each modality and nonzero voxels only.

3.3 Data separation

For training purposes, the training data divided into two major parts, 80% for training and 20% for validation. Data with High-grade Glioma (HGG) and Low-grade Glioma (LGG) category. LGG data from direct examination sometimes does not contain the ET area. To get the best training model, all LGG category data are included in the training. And in order to obtain a more stable training model against variations in the initial weight of the model, the 5-Fold strategy was used.

3.4 Proposed architecture

The proposed architecture is the development of the UNet [9] network with some modifications. The original UNet network was used to process 2D medical images, while currently being faced is 3D medical images with several modalities.This architecture uses multiple dilated convolution and attention blocks to improve the performance of the model. Visualization of this architecture as shown in the Fig. 1.

This architectural form still follows the UNet network pattern but with only one level of downsampling. This architecture also utilizes the power of atrous convolution to enlarge the receptive fields and add attention mechanisms to focus on features relevant to target segmentation. Therefore, this architecture is called Shallow Dilated with Attention UNet 2.5D (SDA-UNet2.5D).

The input is a 3D medical image with size $4 \times 5 \times 196 \times 196$. To be processed with 2D processing, a 3D to 2D converter block is used. This converter block gets a 3D image input and produces a 2D output with a size of $16 \times 196 \times 196$. Details of this block can be seen in Fig. 2. In the 3Dto2D converter, we take five slices of each modality convoluted into eight channels of size $8 \times 5 \times 196 \times 196$. Using these five slices is a 2.5D approximation compared to using all slices in the 3D approximation or only one slice in the 2D approximation. The obtained feature map is then processed with two paths, the first path is a convolution with a kernel size of $5 \times 5 \times 5$, and the second path contains two convolution sequences with a kernel size of $3 \times 3 \times 3$. The use of these two processing lines is intended to capture information from areas with different kernel sizes to enrich each other's output. The output of each path is combined to get a feature map of size $16 \times 1 \times 196 \times 196$, which is then reshaped to size $16 \times 196 \times 196$. The feature map of this converter becomes the input for the next encoder block.

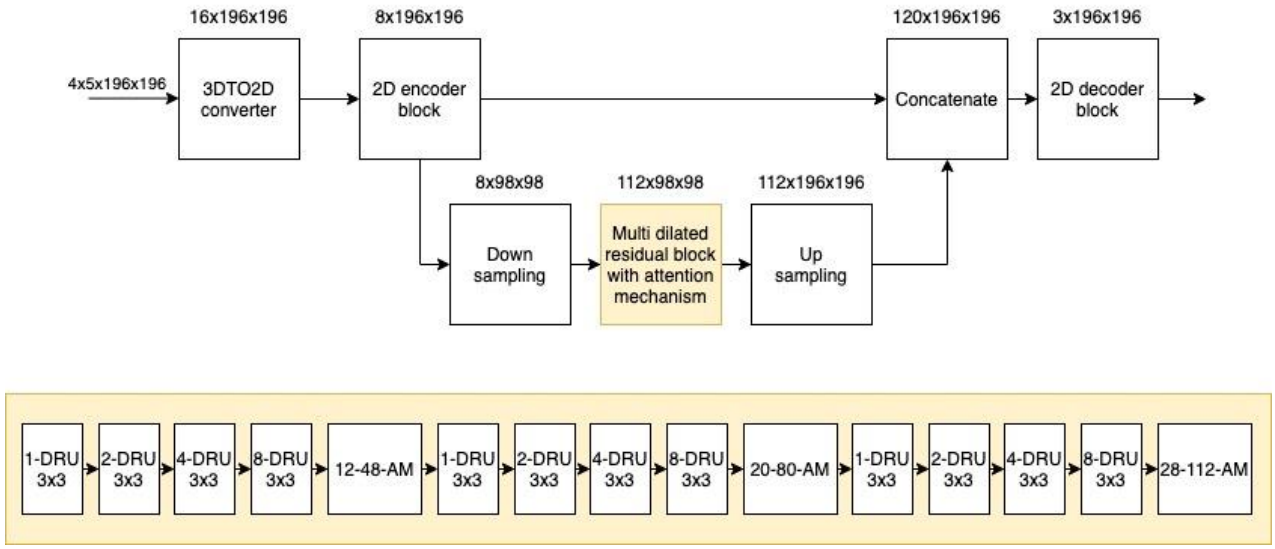

Figure. 1 Shallow dilated with attention UNet 2.5D - SDA-UNet2.5D

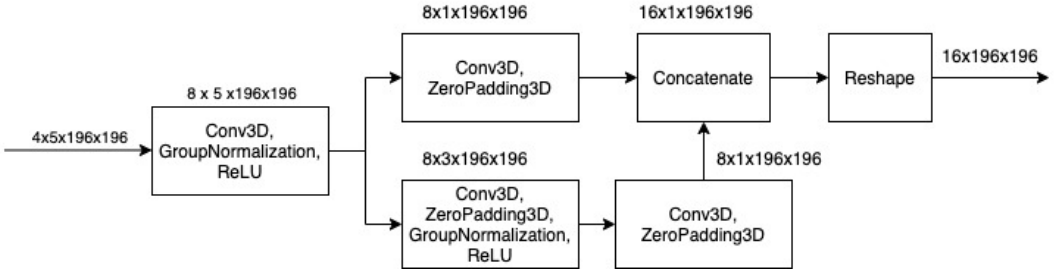

Figure. 2 3D to 2D converter block

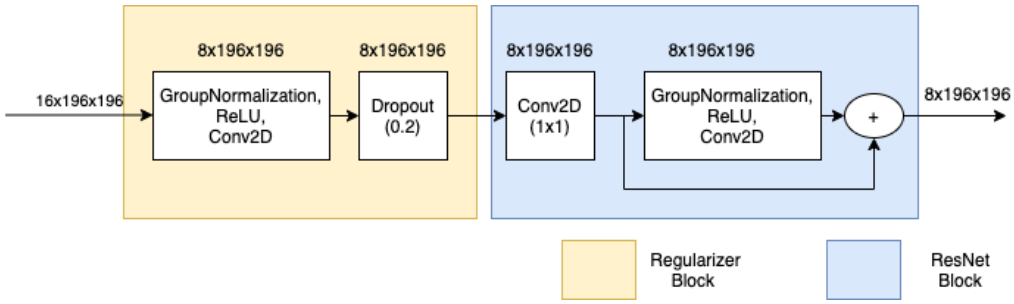

Figure. 3 Encoder block

The 2D Encoder block consists of a regularizer and a residual block. The regularizer block is a convolution block with a kernel size of $3 \times 3 \times 3$, eight filters, and followed by a dropout layer with a rate of 0.2 to prevent overfitting, as suggested by [44]. The selection of rate values for this dropout varies. The value 0.1 is used in [45], 0.5 is used in [46, 47] and [48] finds a rate value of 0.1 from the 0.1-0.5 interval which is the best for their model. Furthermore, the residual block is a ResNet block with a pre-activation strategy. ResNet itself has been proven to be able to accelerate convergence in image recognition as was done in [16]. Meanwhile, alternative activation was investigated in [17] and found that the pre-activation strategy was able to give better results compared to other strategies. This encoder block is structured by adopting these studies. Details of this encoder block can be seen in the Fig. 3.

The first convolution in residual block is used to equalize the number of filters with the output of the second convolution, using the kernel size 1×1. Before proceeding to the subsequent convolution with kernel size 3×3, the feature maps are normalized with GroupNormalization [49] and activated using ReLU. The first convolution feature is added with the second convolution feature to produce block's output.

The encoder block's output is passed as a skip connection to the corresponding layer in the

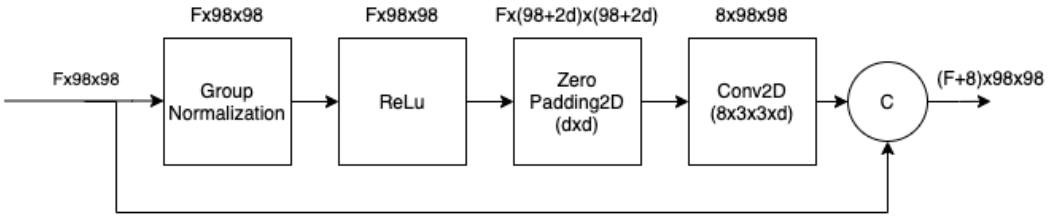

Figure. 4 d-Dilated residual unit

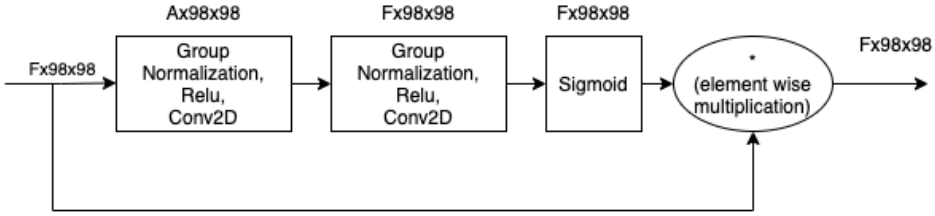

Figure. 5 F-A-Attention mechanisms block

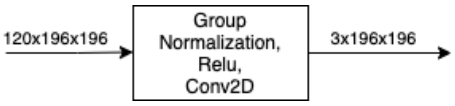

Figure. 6 2D decoder block

Table 3. SDA-UNet2.5D architecture's size

Metrics	Amount
Total Params	167,959
Trainable Params	167,959
Non-trainable Params	0
FLOPS	4.53 G

expanding section. In addition, this output is downsampled using convolution with stride 2×2, kernel size 3×3, and the number of filters is doubled for further processing in the following block.

As a transfer/bottleneck layer, a multi-dilated residual block with attention mechanism is employed as shown in Fig. 1. This multi-dilated residual block with an attention mechanism consists of three sequences of four dilated residual blocks with different dilation factors (d-DRU) followed by an attention mechanism with different filter sizes (F-A-AM). Details of the components that make up d-DRU and F-A-AM as shown in Fig. 4 and 5.

D-DRU consists of some operations starting from GroupNormalization, ReLU, and Conv2D as shown in the Fig. 4. After ReLU, the feature maps are padded using a dilation-valued factor used. After convolution, the feature map will have equal size with the initial feature to be concatenated at the end of the unit.

The F-A-Attention Mechanism is intended to add more focus to a small object area. Enhanced tumor or Necrotic objects from the data generally occupy a small area compared to the Edema area. The diagram of the Attention Unit is as depicted in the Fig. 5. At the end of this attention block, the sigmoid activation result of the last convolution is multiplied by the initial input. The resulting features are passed to the next block.

The block followed is an upsampling block that doubles the feature size by using upsampling. The upsampled features are combined with the skip features of the encoder block to produce features with a size of $120 \times 196 \times 196$, which is the input for the decoder block.

The decoder block, which is the last block in this architecture, is used to process the previous feature maps into segmentation classes being targeted. In this case, the goal of segmentation is three classes, ET, TC, and WT. The visual details of this decoder block are shown in the Fig. 6.

The size of this architecture is light when viewed from the number of parameters and the number of floating-point operations per second (FLOPs) [50] as stated in Table 3.

3.5 Post-processing

The output of this model is three one-hot vectors with dimensions $3 \times 196 \times 196$ representing ET, TC, and WT areas. In order to be validated using the online validation tool, this output must be reconstructed back into a 3D image of size $240 \times 240 \times 155$.

One slice of prediction results from processing five input slices from 4 modalities that are compiled together. To get the prediction results from the image as a whole, all the slices must be processed.

The output transformation of the model results in the form of three one-hot vectors with indexes 0 (ET), 1 (TC), and 2 (WT) to one slice of results is carried out with the following procedure:

1. Duplicate the WT vector to the output vector with values > 0.5 changed to 2, and others changed to 0.
2. Based on the value in the TC vector, change the value in the output vector to 1 if the element in the TC vector is > 0.5.
3. Based on the value in the ET vector, change the value in the output vector to 4 if the element in the ET vector is > 0.5.

The result of the slice transformation is reconstructed back to size $240 \times 240 \times 155$ based on the cropping location.

3.6 Loss function

The dice loss function is used in training and calculated as one minus the dice score. Dice score itself is a function that calculates the intersection between the segmentation results and the existing ground truth. The equation for calculating this dice-score can be seen in Eq. (2), where x represents the area, Y and P represent ground truth and prediction results in one-hot vector form, and ϵ contains small value to avoid dividing by zero.

Because the final output of this architecture is three segmentation classes, it is necessary to find a way to combine them into one value. The ET class score with the smallest area average is given a weight of 0.34, while the TC and WT areas are each given a weight of 0.33. Therefore, the loss function becomes like in Eq. (3).

$$dice_x = \frac{2\times(Y_x \times P_x)+\epsilon}{|Y_x|+|P_x|+\epsilon} \quad (2)$$

$$L_{func} = 1 - (0.34 \times dice_{ET} + 0.33 \times dice_{TC} + 0.33 \times dice_{WT}) \quad (3)$$

Where $dice_{ET}$, $dice_{WT}$, and $dice_{TC}$ are the dice score of each segmented area as shown in the Eq. (2).

3.7 Performance metrics

Segmentation is performed to identify three areas in the image (ET, TC, and WT areas). This measurement is calculated using the Dice Score. The equation to calculate the measurement can be seen in the Eq. (4).

$$dice(P,T) = \frac{2\times TP+\epsilon}{2TP+FP+FN+\epsilon} \times 100\% \quad (4)$$

With P and T representing the prediction and target results, TP and TN represent the number of correctly classified object or background voxels. FN and FP represent the number of incorrectly classified background or object voxels. ϵ is a small number to avoid dividing by zero.

4. Results and discussions

4.1 Experiment settings

The architecture implementation is done by using the Keras/TensorFlow 2.5 package. The hardware used is 64GB of RAM with an Nvidia RTX 2080i 11GB GPU. The batch size used is 8 with one slice per data.

The 3D resolution size used for each data is $196 \times 196 \times 196$; therefore, if the slice size of the 3D image is less than that size, some blank slices will be added. Five slices of the four modalities will be taken with the position of the ground truth slice in the middle. Furthermore, for ground truth slices at indexes 0 and 1, two/one blank slices will be added to each modality previously, respectively. Likewise, two/one blank slices will be added to each modality afterward for ground truth slices at the last two indexes.

Some augmentation techniques were applied to the training data to increase data variations and increase the model's generalizability during training. Augmentation used includes rotation, taking slices based on different axes, replacing one of the modalities with a gaussian distribution, and mirroring.

The rotation technique is performed on 3D images with variations in angles of $-15°$, $0°$, and $15°$. This rotation is performed relative to a randomly selected axis and angle. After the rotation process, several slices and their ground truth were taken to be used in training.

The epoch used is 900. The optimizer used is Adam with a learning rate of 1e-4, and the loss function used is diceloss with the formulation as mentioned in the Eq. (3).

4.2 Results analysis

The time spent on training and local validation in each fold for 900 epochs as stated in Table 4. The table shows that the average time required per fold is 24,545.6 seconds for BraTS 2018 dataset, 28,500.6 seconds for BraTS 2019 dataset, and 31,163.8 seconds for BraTS 2020 dataset. Meanwhile, running the segmentation process on the validation dataset takes an average of 60, 121, and 118.6 seconds per fold for 66 cases in BraTS 2018, 125 cases in BraTS 2019, and 125 cases in BraTS 2020 validation dataset, respectively. Furthermore, using the ensemble from five models, the segmentation time average per case is 4.05, 4.24, and 4.17 seconds in BraTS 2018, BraTS

Table 4. Model training, validation, and inference time

Fold	BraTS 2018		BraTS 2019		BraTS 2020	
	Training 285 cases (s)	Inference 66 cases (s)	Training 336 cases (s)	Inference 125 cases (s)	Training 369 cases (s)	Inference 125 cases (s)
1	24,619	60	28,509	127	31,351	125
2	24,525	60	28,534	119	31,369	117
3	24,515	60	28,513	119	31,345	117
4	24,517	60	28,437	120	30,636	117
5	24,552	60	28,510	120	31,118	117
Average	24,545.6	60	28,500.6	121	31,163.8	118.6
Ensemble 5 models	-	267	-	530	-	521

Table 5. Segmentation time on one patient data

Methods	Segmentation time (seconds)	#params
Pereira et al. [35]	480	2,118K
Havaei et al. [36]	180	802K
Zhao et al. [37]	120-240	-
Hu et al. [7]	90-180	-
Chen et al.[38]	186.93	100K
SDA-UNet2.5D	4.05-4.24	168K

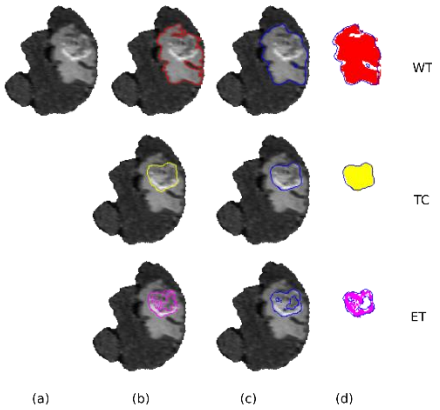

Figure. 7 Segmentation results on the 80th slice: (a) flair image on the 80th slice, (b) image with ground truth boundary, (c) image with prediction boundary, and (d) overlapping boundary of ground truth and prediction

2019, and BraTS 2020 validation dataset. This performance can be achieved with the model's size, which only has 167,959 trained parameters with a FLOPs size of 4.53 GFlops.

Comparing the segmentation speed with a number of the latest methods, the segmentation speed of this architecture provides the best performance as shown in Table 5. The DF MLDeepMedic [38] architecture has fewer parameters than the proposed architecture, but the segmentation speed is still below the proposed architecture. This is possible because DF MLDeepMedic uses additional postprocessing by using the multi-layer perceptron (MLP) and small image size pieces. In contrast, this proposed architecture uses simple postprocessing by directly converting the final output into individual tumor parts and compiling them into one output. In addition, using a small chunk size ($25 \times 25 \times 25$) in DF MLDeepMedic causes more processing steps to be done when compared to using a larger chunk as used by the proposed architecture ($5 \times 196 \times 196$). The use of small image size pieces is also carried out in [7, 35-37], which causes iterative processing to process the complete image so that each architecture requires a longer inference time than the proposed architecture. Using a fully connected layer in [35] and additional postprocessing in [37] causes the inference time to be longer than the proposed architecture.

Visual analysis of the segmentation results using the fold 1 model as shown in Fig. 7. The image is Flair's modality at the 80th slice. The first row is a visualization for the WT area, the second row for the TC area, and the third row for the ET area.

There is a curve in ground truth in the WT area that is slightly recognizable by the prediction results. Likewise, ET areas with complex ground truth areas can still be recognized in predictions but with a more general area shape. However, for the TC area, the prediction results and ground truth intersect well.

As seen in the fourth column, the overlap between the segmentation result area and ground truth is mainly in the middle. Meanwhile, there is a discrepancy between the prediction area and the ground truth in the edge area. There are parts of the segmentation results that are more protruding than the ground truth, and there is also the area of the segmentation results that is smaller than the ground truth.

With the 5-fold cross-validation training strategy, five different trained weights were produced. Each of these weights is used to segment the validation dataset. In addition, the ensembling of 5 model weights to segment the validation dataset using the averaging method was also carried out. The results of each segmentation are sent to an online validation tool to check for accuracy. Table 6 to 8 shows the performance of each model and the combination of the five models on each validation datasets.

In the BraTS 2018 validation dataset, the performance of the proposed architecture is superior, with the dice performance for the ensemble model reaching 75.70, 77.33, and 88.82 for the ET, TC, and WT areas. The variation in the performance of each 5-fold training model is also not too high. Furthermore, combining five models of each fold increase the average performance of the dice. The number of data validated online is 66 patient data. The time required to perform inference is 267 seconds for the ensemble model, so the average time required per patient data is 4.05 seconds.

In the BraTS 2019 validation dataset, the proposed architecture also provides good dice average performance. The model used is a model that was trained from the beginning for each fold, so it does not use the model from the previous training. The resulting performance is still good, although slightly decreased compared to the architectural performance in the BraTS 2018 validation dataset. This variation is acceptable given the variations in the dataset that may occur and the training data used. As previously mentioned, the 2019 BraTS validation dataset consists of 125 patient data with 336 training data.

For architectural performance in the BraTS 2020 validation dataset, the average dice for each area also gives good results. Although it decreased compared to the performance in the previous validation dataset, the architectural performance was still good in this BraTS 2020 dataset.

To determine the quality of the architecture, we also compare the performance of the proposed architecture with several current architectures for each validation dataset. Architectural performance comparison across the BraTS 2018 validation dataset as shown in Table 9.

The average dice performance of the proposed architecture in the BraTS 2018 validation dataset outperformed all comparison architectures in all areas. The 2D processing approach for 3D MRI images such as that of Tuan et al. [6] and Benson et al. [26] has lower performance than the proposed architecture using the 2.5D approach. Tuan et al. [6], which uses two types of kernel sizes (3×3 and 5×5), still performs lower performance than the proposed architecture that utilizes atrous convolution to expand the receptive area. Likewise, the approach was taken by Hu et al. [7] which combines the 2D approach of three axes also has lower performance. The proposed architecture is still superior to The GAN architectural approach with 3D processing carried out by Rezaei et al. [31]. Only in the TC section, the proposed architecture is lower than the architecture in Rezaei et al. [8] but has higher performance in ET and WT areas. These results show that using the 2.5D approach with atrous convolution in the proposed architecture can outperform the comparison architecture that uses 2D processing and 3D processing approaches by utilizing GAN for the 3D MRI images.

In the BraTS 2019 validation dataset as seen in Table. 10, the average performance of the proposed architecture dice also outperforms all other comparison architectures. The proposed architecture outperforms the architecture in [27] and [14] which use a 2D approach. The proposed architecture also

Table 6. Segmentation performance on BraTS 2018 Validation dataset

Fold	Mean Dice (%)		
	ET	WT	TC
1	72.83	87.53	76.55
2	74.36	87.55	76.25
3	71.29	85.15	75.19
4	71.49	87.05	75.21
5	72.95	87.48	73.13
Ensemble	75.70	88.82	77.33

Table 7. Segmentation performance on BraTS 2019 Validation dataset

Fold	Mean Dice (%)		
	ET	WT	TC
1	69.61	85.32	75.39
2	68.68	85.40	74.58
3	69.84	86.29	75.24
4	66.79	86.57	74.08
5	69.34	87.10	75.09
Ensemble	71.29	88.00	76.55

Table 8. Segmentation performance on BraTS 2020 Validation dataset

Fold	Mean Dice (%)		
	ET	WT	TC
1	66.76	85.97	74.08
2	69.95	86.90	74.05
3	68.95	85.59	73.64
4	66.42	85.50	74.58
5	68.09	86.48	74.69
Ensemble	70.80	87.95	75.89

Table 9. Dice performance comparation on the BraTS 2018 validation dataset

Arch.	Mean Dice (%)		
	ET	WT	TC
Tuan et al.[6]	68.25	81.87	69.99
Rezaei et al.[8]	63	84	79
Rezaei et al. [31]	61	81	64
Benson et al. [26]	66	82	72
Hu et al.[7]	71.78	88.24	74.81
SDA-UNET2.5D	75.70	88.82	77.33

Table 10. Dice performance comparation on the BraTS 2019 validation dataset

Arch.	Mean Dice (%)		
	ET	WT	TC
Guo et al. [42]	67.7	87.2	72.8
Lorenzo et al.[14]	66.34	89.04	75.11
Ahmad et al. [29]	62.30	85.18	75.76
Kotowski et al.[27]	68.4	83.8	73.5
Bhalerao et al. [28]	66.68	85.27	70.91
Yan et al.[43]	66	86	73
SDA-UNET2.5D	71.29	88.00	76.55

Table 11. Dice performance comparation on the BraTS 2020 validation dataset

Arch.	Mean Dice (%)		
	ET	WT	TC
Awasthi et al.[39]	57	73	61
Savadikar et al.[41]	68.89	81.90	71.68
Zhao et al.[33]	67.1	86.2	62.3
Agravat et al.[32]	68.6	87.6	72.5
Kotowski et al.[30]	68.53	87.12	74.53
Xu et al. [40]	67.36	86.08	70.42
SDA-UNET2.5D	70.80	87.95	75.89

outperforms the dice performance of architectures using a 3D approach such as in [28, 29]. The proposed architecture also outperforms the architecture in [42] which uses the attention mechanism, and [43], which uses SEB in the architecture. The use of Multi-dilated block and attention mechanism in the proposed architecture provides dice performance that outperforms the use of atrous convolution in [29], variation of attention mechanism in [42], and SEB block in [43].

The combined use of several atrous convolutions with different dilatation factors and attention mechanisms in one block makes the proposed architecture outperform the comparison architecture that uses the attention mechanism in the skip connection section in [39] as shown in Table 11. The proposed architecture also outperforms the architecture in [41], which uses probabilistic UNet by applying an attention mechanism to the skip connection. The performance of the architecture in [40] which adds deep supervision in the decoder section, and the architecture in [33] that uses the SEB block also has a lower average dice performance than the proposed architecture.

5. Conclusion and future works

We have proposed the Shallow Dilated with Attention-UNet2.5D architecture, which is a development of the UNet architecture with a 2.5D approach, consists of only one level of downsampling, and contains a Multi Dilated Residual with Attention Mechanism block in the transfer section. The architecture was tested using the BraTS 2018, BraTS 2019, and BraTS 2020 datasets and yielded superior performance over the current benchmark architectures. The advantages of this architecture are supported by one level of downsampling in UNet, the use of atrous convolutions with different dilatation factors, and attentional mechanisms in the Multi-dilated residual Attention mechanism block. Atrous convolutions increase the receptive area, attentional mechanisms amplify the relevant features added, and residual pathways prevent the architecture from being vanishing gradient even though these blocks are arranged in many layers. The proposed architecture also has a superior inference speed than other comparison architectures, with an average inference speed of 4.05 - 4.24 seconds per patient data.

One concern from this proposed architecture is the results of validation across three datasets. Although the online validation results show superior results, the dice segmentation performance decreased along with the dataset used from the BraTS 2018 validation dataset, BraTS 2019, and BraTS 2020. For example, the average performance of the ET dice area for the ensemble model resulted in 75.70 in the BraTS 2018 validation dataset, decreased to 71.29 in the BraTS 2019 validation dataset, and finally decreased to 70.80 for the BraTS 2020 validation dataset. The training data for each dataset is 286, 336, and 369 patient data for BraTS 2018, BraTS 2019, and BraTS 2020. In comparison, the number of validation data includes 66, 125, and 125 for BraTS 2018, BraTS 2019, and BraTS 2020, respectively. This declining performance needs further study.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Agus Subhan Akbar, Chastine Fatichah, and Nanik Suciati; methodology, Agus Subhan Akbar, Chastine Fatichah and Nanik Suciati;

software, Agus Subhan Akbar; validation, Agus Subhan Akbar; formal analysis, Agus Subhan Akbar, Chastine Fatichah and Nanik Suciati; investigation, Agus Subhan Akbar, Chastine Fatichah and Nanik Suciati; resources, Agus Subhan Akbar, Chastine Fatichah and Nanik Suciati; writing—original draft preparation, Agus Subhan Akbar; writing—review and editing, Agus Subhan Akbar, Chastine Fatichah and Nanik Suciati; visualization, Agus Subhan Akbar; supervision, Chastine Fatichah and Nanik Suciati.

Acknowledgments

This work was supported by the Ministry of Education and Culture, Indonesia. We are deeply grateful for BPPDN (Beasiswa Pendidikan Pascasarjana Dalam Negeri) and PDD (Penelitian Disertasi Doktor) 2020-2021 under Grant No. 003/E4/AK.04.PTNBH/2021, which enabled this research could be done.

References

[1] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection", *Nat Sci Data*, Vol. 4, p. 170117, 2017.

[2] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection", *Nat Sci Data*, Vol. 4, p. 170117, 2017.

[3] B. H. Menze, A. Jakab, S. Bauer, J. K. Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. V. Leemput, "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)", *IEEE Transactions on Medical Imaging*, Vol. 34, No. 10, pp. 1993-2024, 2015.

[4] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", *Scientific Data*, Vol. 4, p. 170117, 2017.

[5] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, M. Prastawa, E. Alberts, J. Lipkova, J. Freymann, J. Kirby, M. Bilello, H. F. Shaykh, R. Wiest, J. Kirschke, B. Wiestler, R. Colen, A. Kotrotsou, P. Lamontagne, D. Marcus, M. Milchenko, A. Nazeri, M. A. Weber, A. Mahajan, U. Baid, E. Gerstner, D. Kwon, G. Acharya, M. Agarwal, M. Alam, A. Albiol, A. Albiol, F. J. Albiol, V. Alex, N. Allinson, P. H. A. Amorim, A. Amrutkar, G. Anand, S. Andermatt, T. Arbel, P. Arbelaez, A. Avery, M. Azmat, P. B., W. Bai, S. Banerjee, B. Barth, T. Batchelder, K. Batmanghelich, E. Battistella, A. Beers, M. Belyaev, M. Bendszus, E. Benson, J. Bernal, H. N. Bharath, G. Biros, S. Bisdas, J. Brown, M. Cabezas, S. Cao, J. M. Cardoso, E. N. Carver, A. Casamitjana, L. S. Castillo, M. Catà, P. Cattin, A. Cerigues, V. S. Chagas, S. Chandra, Y. J. Chang, S. Chang, K. Chang, J. Chazalon, S. Chen, W. Chen, J. W. Chen, Z. Chen, K. Cheng, A. R. Choudhury, R. Chylla, A. Clérigues, S. Colleman, R. G. R. Colmeiro, M. Combalia, A. Costa, X. Cui, Z. Dai, L. Dai, L. A. Daza, E. Deutsch, C. Ding, C. Dong, S. Dong, W. Dudzik, Z. E. Rosen, G. Egan, G. Escudero, T. Estienne, R. Everson, J. Fabrizio, Y. Fan, L. Fang, X. Feng, E. Ferrante, L. Fidon, M. Fischer, A. P. French, N. Fridman, H. Fu, D. Fuentes, Y. Gao, E. Gates, D. Gering, A. Gholami, W. Gierke, B. Glocker, M. Gong, S. G. Villá, T. Grosges, Y. Guan, S. Guo, S. Gupta, W. S. Han, I. S. Han, K. Harmuth, H. He, A. H. Sabaté, E. Herrmann, N. Himthani, W. Hsu, C. Hsu, X. Hu, X. Hu, Y. Hu, Y. Hu, R. Hua, T. Y. Huang, W. Huang, S. V. Huffel, Q. Huo, V. Hv, K. M. Iftekharuddin, F. Isensee, M. Islam, A. S. Jackson, S. R. Jambawalikar, A. Jesson, W. Jian, P. Jin, V. J. M. Jose, A. Jungo, B. Kainz, K. Kamnitsas, P. Y. Kao, A. Karnawat, T. Kellermeier, A. Kermi, K. Keutzer, M. T. Khadir, M. Khened, P. Kickingereder, G. Kim, N. King, H. Knapp, U. Knecht, L. Kohli, D. Kong, X. Kong, S. Koppers, A. Kori, G. Krishnamurthi, E. Krivov, P. Kumar, K. Kushibar, D. Lachinov, T. Lambrou, J. Lee, C. Lee, Y. Lee, M. Lee, S. Lefkovits, L. Lefkovits, J. Levitt, T. Li, H. Li, W. Li, H. Li, X. Li, Y. Li, H. Li, Z. Li, X. Li, Z. Li, X. Li, W. Li, Z. S. Lin,

F. Lin, P. Lio, C. Liu, B. Liu, X. Liu, M. Liu, J. Liu, L. Liu, X. Llado, M. M. Lopez, P. R. Lorenzo, Z. Lu, L. Luo, Z. Luo, J. Ma, K. Ma, T. Mackie, A. Madabushi, I. Mahmoudi, K. H. M. Hein, P. Maji, C. Mammen, A. Mang, B. S. Manjunath, M. Marcinkiewicz, S. McDonagh, S. McKenna, R. McKinley, M. Mehl, S. Mehta, R. Mehta, R. Meier, C. Meinel, D. Merhof, C. Meyer, R. Miller, S. Mitra, A. Moiyadi, D. M. Garcia, M. A. B. Monteiro, G. Mrukwa, A. Myronenko, J. Nalepa, T. Ngo, D. Nie, H. Ning, C. Niu, N. K. Nuechterlein, E. Oermann, A. Oliveira, D. D. C. Oliveira, A. Oliver, A. F. I. Osman, Y. N. Ou, S. Ourselin, N. Paragios, M. S. Park, B. Paschke, J. G. Pauloski, K. Pawar, N. Pawlowski, L. Pei, S. Peng, S. M. Pereira, J. P. Beteta, V. M. P. Garcia, S. Pezold, B. Pham, A. Phophalia, G. Piella, G. N. Pillai, M. Piraud, M. Pisov, A. Popli, M. P. Pound, R. Pourreza, P. Prasanna, V. Prkovska, T. P. Pridmore, S. Puch, É. Puybareau, B. Qian, X. Qiao, M. Rajchl, S. Rane, M. Rebsamen, H. Ren, X. Ren, K. Revanuru, M. Rezaei, O. Rippel, L. C. Rivera, C. Robert, B. Rosen, D. Rueckert, M. Safwan, M. Salem, J. Salvi, I. Sanchez, I. Sánchez, H. M. Santos, E. Sartor, D. Schellingerhout, K. Scheufele, M. R. Scott, A. A. Scussel, S. Sedlar, J. P. S. Rubio, N. J. Shah, N. Shah, M. Shaikh, B. U. Shankar, Z. Shboul, H. Shen, D. Shen, L. Shen, H. Shen, V. Shenoy, F. Shi, H. E. Shin, H. Shu, D. Sima, M. Sinclair, O. Smedby, J. M. Snyder, M. Soltaninejad, G. Song, M. Soni, J. Stawiaski, S. Subramanian, L. Sun, R. Sun, J. Sun, K. Sun, Y. Sun, G. Sun, S. Sun, Y. R. Suter, L. Szilagyi, S. Talbar, D. Tao, D. Tao, Z. Teng, S. Thakur, M. H. Thakur, S. Tharakan, P. Tiwari, G. Tochon, T. Tran, Y. M. Tsai, K. L. Tseng, T. A. Tuan, V. Turlapov, N. Tustison, M. Vakalopoulou, S. Valverde, R. Vanguri, E. Vasiliev, J. Ventura, L. Vera, T. Vercauteren, C. A. Verrastro, L. Vidyaratne, V. Vilaplana, A. Vivekanandan, G. Wang, Q. Wang, C. J. Wang, W. Wang, D. Wang, R. Wang, Y. Wang, C. Wang, G. Wang, N. Wen, X. Wen, L. Weninger, W. Wick, S. Wu, Q. Wu, Y. Wu, Y. Xia, Y. Xu, X. Xu, P. Xu, T. L. Yang, X. Yang, H. Y. Yang, J. Yang, H. Yang, G. Yang, H. Yao, X. Ye, C. Yin, B. Y. Moxon, J. Yu, X. Yue, S. Zhang, A. Zhang, K. Zhang, X. Zhang, L. Zhang, X. Zhang, Y. Zhang, L. Zhang, J. Zhang, X. Zhang, T. Zhang, S. Zhao, Y. Zhao, X. Zhao, L. Zhao, Y. Zheng, L. Zhong, C. Zhou, X. Zhou, F. Zhou, H. Zhu, J. Zhu, Y. Zhuge, W. Zong, J. K. Cramer, K. Farahani, C. Davatzikos, K. V. Leemput, and B. Menze, "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge", 2018.

[6] T. A. Tuan, T. A. Tuan, and P. T. Bao, "Brain Tumor Segmentation Using Bit-plane and UNET", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 466-475, 2019.

[7] K. Hu, Q. Gan, Y. Zhang, S. Deng, F. Xiao, W. Huang, C. Cao, and X. Gao, "Brain Tumor Segmentation Using Multi-Cascaded Convolutional Neural Networks and Conditional Random Field", *IEEE Access*, Vol. 7, pp. 92615-92629, 2019.

[8] M. Rezaei, H. Yang, and C. Meinel, "voxel-GAN: Adversarial Framework for Learning Imbalanced Brain Tumor Segmentation", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 321-333, 2019.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", In: *Proc. of Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, pp. 234-241, 2015.

[10] M. Aghalari, A. Aghagolzadeh, and M. Ezoji, "Brain tumor image segmentation via asymmetric/symmetric UNet based on two-pathway-residual blocks", *Biomedical Signal Processing and Control*, Vol. 69, p. 102841, 2021.

[11] L. Han, Y. Chen, J. Li, B. Zhong, Y. Lei, and M. Sun, "Liver segmentation with 2.5D perpendicular UNets", *Computers & Electrical Engineering*, Vol. 91, p. 107118, 2021.

[12] S. V. and I. G., "Encoder Enhanced Atrous (EEA) Unet architecture for Retinal Blood vessel segmentation", *Cognitive Systems Research*, Vol. 67, pp. 84-95, 2021.

[13] V. T. Pham, T. T. Tran, P. C. Wang, P. Y. Chen, and M. T. Lo, "EAR-UNet: A deep learning-based approach for segmentation of tympanic membranes from otoscopic images", *Artificial Intelligence in Medicine*, Vol. 115, p. 102065, 2021.

[14] P. R. Lorenzo, M. Marcinkiewicz, and J. Nalepa, "Multi-modal U-Nets with Boundary Loss and Pre-training for Brain Tumor Segmentation", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 135-147, 2020.

[15] R. Ge, H. Cai, X. Yuan, F. Qin, Y. Huang, P. Wang, and L. Lyu, "{MD}-{UNET}: Multi-input dilated U-shape neural network for

segmentation of bladder cancer", *Computational Biology and Chemistry*, Vol. 93, p. 107510, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks", In: *Proc. of Computer Vision -- ECCV 2016*, pp. 630-645, 2016.

[18] A. Araujo, W. Norris, and J. Sim, "Computing Receptive Fields of Convolutional Neural Networks", *Distill*, Vol. 4, No. 11, 2019.

[19] N. Adaloglou, "Understanding the receptive field of deep convolutional networks", *https://theaisummer.com/receptive-field/*, 2020, Accessed: Aug. 01, 2021. [Online]. Available: https://theaisummer.com/receptive-field/

[20] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation", *ArXiv*, 2017.

[21] O. Oktay, J. Schlemper, L. L. 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", *ArXiv*, 2018.

[22] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images", *Medical Image Analysis*, Vol. 53, pp. 197-207, 2019.

[23] K. Trebing, T. Stańczyk, and S. Mehrkanoon, "SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture", *Pattern Recognition Letters*, Vol. 145, pp. 178-186, 2021.

[24] H. P. A. Tjahyaningtijas, D. J. Rumala, C. V. Angkoso, N. Z. Fanani, J. Santoso, A. D. Sensusiati, P. M. A. V. Ooijen, I. K. E. Purnama, and M. Purnomo, "Brain Tumor Classification in MRI Images Using En-CNN", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 437-451, 2021.

[25] S. K. V. Rao and B. Lingappa, "Image Analysis for MRI Based Brain Tumour Detection Using Hybrid Segmentation and Deep Learning Classification Technique", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 5, pp. 53-62, 2019.

[26] E. Benson, M. P. Pound, A. P. French, A. S. Jackson, and T. P. Pridmore, "Deep Hourglass for Brain Tumor Segmentation", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 419-428, 2019.

[27] K. Kotowski, J. Nalepa, and W. Dudzik, "Detection and Segmentation of Brain Tumors from MRI Using U-Nets", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 179-190, 2020.

[28] M. Bhalerao and S. Thakur, "Brain Tumor Segmentation Based on 3D Residual U-Net", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 218-225, 2020.

[29] P. Ahmad, S. Qamar, S. R. Hashemi, and L. Shen, "Hybrid Labels for Brain Tumor Segmentation", *Springer International Publishing*, pp. 158-166, 2020.

[30] K. Kotowski, S. Adamski, W. Malara, B. Machura, L. Zarudzki, and J. Nalepa, "Segmenting Brain Tumors from MRI Using Cascaded 3D U-Nets", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 265-277, 2021.

[31] M. Rezaei, K. Harmuth, W. Gierke, T. Kellermeier, M. Fischer, H. Yang, and C. Meinel, "A Conditional Adversarial Network for Semantic Segmentation of Brain Tumor", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 241-252, 2018.

[32] R. R. Agravat and M. S. Raval, "3D Semantic Segmentation of Brain Tumor for Overall Survival Prediction", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 215-227, 2021.

[33] C. Zhao, Z. Zhao, Q. Zeng, and Y. Feng, "MVP U-Net: Multi-View Pointwise U-Net for Brain Tumor Segmentation", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 93-103, 2021.

[34] R. Indraswari, T. Kurita, A. Z. Arifin, N. Suciati, and E. R. Astuti, "Multi-projection deep learning network for segmentation of 3D medical images", *Pattern Recognition Letters*, Vol. 125, pp. 791-797, 2019.

[35] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images", *IEEE Transactions on Medical Imaging*, Vol. 35, No. 5, pp. 1240-1251, 2016.

[36] M. Havaei, A. Davy, D. W. Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P. M. Jodoin, and H. Larochelle, "Brain tumor segmentation with Deep Neural Networks", *Medical Image Analysis*, Vol. 35, pp. 18-31, 2017.

[37] X. Zhao, Y. Wu, G. Song, Z. Li, Y. Zhang, and Y. Fan, "A deep learning model integrating FCNNs and CRFs for brain tumor segmentation", *Medical Image Analysis*, Vol. 43, pp. 98-111, 2018.

[38] S. Chen, C. Ding, and M. Liu, "Dual-force convolutional neural networks for accurate brain tumor segmentation", *Pattern Recognition*, Vol. 88, pp. 90-100, 2019.

[39] N. Awasthi, R. Pardasani, and S. Gupta, "Multi-threshold Attention U-Net (MTAU) Based Model for Multimodal Brain Tumor Segmentation in MRI Scans", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 168-178, 2021.

[40] J. H. Xu, W. P. K. Teng, X. J. Wang, and A. Nürnberger, "A Deep Supervised U-Attention Net for Pixel-Wise Brain Tumor Segmentation", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 278-289, 2021.

[41] C. Savadikar, R. Kulhalli, and B. Garware, "Brain Tumour Segmentation Using Probabilistic U-Net", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 12659 LNCS, pp. 255-264, 2021.

[42] X. Guo, C. Yang, T. Ma, P. Zhou, S. Lu, N. Ji, D. Li, T. Wang, and H. Lv, "Brain Tumor Segmentation Based on Attention Mechanism and Multi-model Fusion", 2020.

[43] K. Yan, Q. Sun, L. Li, and Z. Li, "3D Deep Residual Encoder-Decoder CNNS with Squeeze-and-Excitation for Brain Tumor Segmentation", In: *Proc. of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 234-243, 2020.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929-1958, 2014.

[45] T. Yang, J. Song, and L. Li, "A deep learning model integrating SK-TPCNN and random forests for brain tumor segmentation in MRI", *Biocybernetics and Biomedical Engineering*, Vol. 39, No. 3, pp. 613-623, 2019.

[46] H. Xie, D. Yang, N. Sun, Z. Chen, and Y. Zhang, "Automated pulmonary nodule detection in CT images using deep convolutional neural networks", *Pattern Recognition*, Vol. 85, pp. 109-119, 2019.

[47] J. Chang, L. Zhang, N. Gu, X. Zhang, M. Ye, R. Yin, and Q. Meng, "A mix-pooling CNN architecture with FCRF for brain tumor segmentation", *Journal of Visual Communication and Image Representation*, Vol. 58, pp. 316-322, 2019.

[48] L. Liu, F. X. Wu, and J. Wang, "Efficient multi-kernel DCNN with pixel dropout for stroke MRI segmentation", *Neurocomputing*, Vol. 350, pp. 117-127, 2019.

[49] Y. Wu and K. He, "Group Normalization", *International Journal of Computer Vision*, Vol. 128, No. 3, pp. 742-755, 2020.

[50] T. Tokusumi, "Keras-Flops - PyPI", *https://pypi.org/project/keras-flops/*, 2020, Accessed: Aug. 01, 2021. [Online]. Available: https://pypi.org/project/keras-flops/