3D S-UNET: an Efficient Architecture for 3 Dimensional Segmentation of Brain Tumors on MRI Images

M Sadewa Wicaksana Wibowo [a,1,*], Muhammad Shodiq [a,2], Bety Qorry Aina [a,3], Angga Lisdiyanto[b,4]

[a] University of Muhammadiyah Lamongan, East Java, Lamongan, Indonesia
[b] University of Pembangunan Nasional "Veteran" East Java, East Java, Surabaya, Indonesia
[1] wicaksanasadewa@yahoo.com*; [2] shodiqmuhammad13@gmail.com; [3] betyaina246@gmail.com; [4] angga_lisdiyanto.mti@upnjatim.ac.id
* corresponding author

ARTICLE INFO

Keywords
Brain Cancer
Deep Learning
Lightweight Deep Learning
Brain Tumor Segmentation 3D

ABSTRACT

One of the deadliest diseases worldwide is brain tumors. In identifying brain tumors, experts perform a subjective analysis that requires considerable time. Previous research has developed automatic 3D brain tumor segmentation using Deep Learning (DL) approaches such as 3D UNet and 3D ResNet. However, these approaches demand significant computational resources. In resource-constrained settings, key criteria for determining the best architecture include memory consumption, inference speed, and accuracy. Therefore, this study introduces the development of the 3D S-UNet architecture, constructed by combining 3D ShuffleNet-V2 as an encoder and 3D UNet as a decoder. The integration of these 3D data processors allows the architecture to be more precise in identifying brain tumor locations and capture richer feature values compared to 2D data processing. The researchers compare 3D S-UNet with 3D U-Net and another Lightweight Deep Learning architecture, 3D Mobile-UNet. The results show that 3D S-UNet has a smaller memory consumption, using 0.56GB for the highest allocated memory and 1.71GB for reserved memory. In terms of inference speed, 3D S-UNet is faster compared to the other three architectures, achieving a speed of 135.881 milliseconds. 3D S-UNet demonstrates favorable results with a Whole Tumor (WT) dice score, sensitivity, and specificity of 83%, 85%, and 88%, respectively.

1. Introduction

The incidence of brain tumors has been increasing each year, particularly in Southeast Asia and Europe [1]. This disease happens when abnormal growths of cells within the skull can be either cancerous or noncancerous but are often serious health issues or even cause death [2]. The diagnostic segmentation results supplied by medical specialists are crucial in choosing treatment plans. Common Therapeutic Approaches Radiotherapy and chemotherapy are common therapeutic techniques, but both have severe side effects, leaving the ideal treatment option up to debate. Some experts advise for maximal surgical resection as both a therapeutic and diagnostic measure, while others suggest postponing surgery for additional examination [3]. To ensure the results from different medical experts can more accurate and help it to analyze the brain tumor some researchers tried to build an automated detection using artificial intelligence to analyze complex abnormal patterns is increasingly necessary [4].

Deep Learning (DL), a subfield of machine learning, uses deep neural network designs to analyze and understand complex data like images, audio, text, and other modalities. DL models are commonly used in medical imaging for four key tasks: classification, object localization, object detection, and segmentation. They typically have several network layers [5]. In medical image segmentation, deep learning-based algorithms often process picture slices or full volumes to identify specific features [6] [7]. Segmentation provides critical information on organ shape and volume and typically involves four stages: dataset partitioning into training, validation, and testing sets; preprocessing (including augmentation through scaling and rotation to enhance dataset diversity); model training; and performance evaluation. Kamnitsas et al. [8] developed DeepMedic, a 3D CNN model capable of exploiting more spatial information while processing the entire volumetric data. Given the success of 3D segmentation, new architectures have emerged. Beers et al. [9] presented a 3D U-Net for volumetric MRI inputs of size 4×32×32×32. Others improved U-Net performance with a dice coefficient loss function to solve class imbalance and data augmentation to reduce overfitting [10]. Myronenko et al. [11] improved 3D segmentation by including autoencoder regularization, whereas Xu et al. [12] used a Deep Cascaded Attention Network based on UNet. Despite their excellent accuracy, many of these models need significant processing power and memory, rendering them unsuitable for resource-constrained applications.

As a result, lightweight deep learning, defined as deep learning models with decreased complexity, fewer parameters, and lower computing needs, can successfully solve issues such as limited processor power, memory capacity, and resource limits. These versions are designed to maintain excellent performance while reducing hardware needs, making them suited for use in contexts such as mobile devices, embedded systems, and edge computing platforms [13][14][15]. One of the lightweight deep learning is Shuffle-Net there are a lot of previous researchers which used Shuffle-Net to implement as an efficient architecture on medical image problems, Chattopadhyay [16] et al build DRDA-Net as dense residual dual-shuffle attention network for breast cancer. Dipankar et al [17] build an architecture CDAM-Net as channel shuffle dual attention based multi-scale CNN for efficient glaucoma detection, and Namho et al [18] created a Shuffled ECA-Net for stress detection from multimodal wearable sensor data. ShuffleNet is a form of convolutional neural network created primarily for devices with minimal computing capability, such as smartphones or embedded systems. Its primary purpose is to maintain the highest level of accuracy while lowering the amount of calculations required by the network. The central notion is derived from two techniques: pointwise group convolution and channel shuffle. 1×1 convolutions, which combine input from many channels, account for a significant percentage of the computational cost in CNNs. ShuffleNet decreases costs by dividing channels into groups and performing 1×1 convolution to each group individually. This "group convolution" approach greatly decreases the amount of multiplications, but it also introduces a problem: channels inside different groups cannot easily interchange information. In this paper, we propose a 3D S-U-Net architecture that combines 3D ShuffleNetV2—a lightweight DL encoder—with a 3D U-Net decoder to achieve accurate brain tumor segmentation while minimizing computational and memory demands. We compared 3D S-UNet to 3D U-Net and 3D Mobile-UNet, which are two different deep learning models. The references of the architecture 3D UNet are the same with the sangui et al which implement the standard 3D UNet [19] and Pendse et al [20] introduce 3D Mobile-UNet with the propose method for improving the 3D U-Net architecture for brain tumor segmentation with the key innovation is the combination of memory-saving techniques to handle the large data volume and memory demands of 3D medical imaging with the based architecture is MobileNet V2.

2. Method

2.1. Methodological Frameworks

Figure 1 depicts the study's methodological framework, which details the sequential procedure used to construct the suggested 3D S-UNet architecture for brain tumor segmentation. The workflow depicts the stages involved in conducting this study.

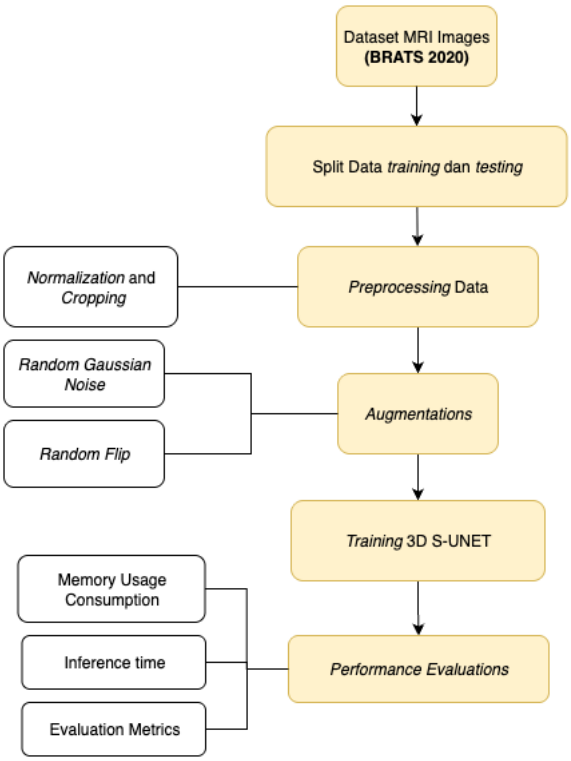

Figure 1 Workflow Research

2.2. Data Collection

The dataset used in this study is BRATS 2020, obtained from the CBICA — The Center for Biomedical Image Computing and Analytics Image Processing Portal [21], [22]. This dataset contains survival data and name mapping, which are utilized to provide additional information for each sample in the dataset. It consists of 369 subjects, including 293 HGG (High Grade Glioma) cases and 76 LGG (Low Grade Glioma) instances. HGG refers to malignant brain tumours that spread rapidly, whereas LGG refers to benign brain tumours that spread slowly. Each data sample includes five images: four brain tumor modality images (*T1, T1-ce, T2,* and *Flair*) and one ground truth image, all with dimensions of 240×240×155. The explanations for each modality are: *T1-Weighted* It is a modality that can discriminate between fat and water. *T1-Contrast-Enhanced (T1-ce)* It is widely utilized to provide a sharper image of locations with increased blood supply or vascular anomalies, such as malignancies. *T2-weighted* It can improve the distinction between various types of water-containing tissues. *Fluid-attenuated inversion recovery (FLAIR)* It is used to distinguish between lesions within or surrounding the brain and CSF, which are not usually evident in typical T1 or T2 images. Each image in this dataset is in the.nii.gz or NIfTI format, which is a medical image format for storing data and embedded information. The illustration of each modality can be seen on Figure 2.

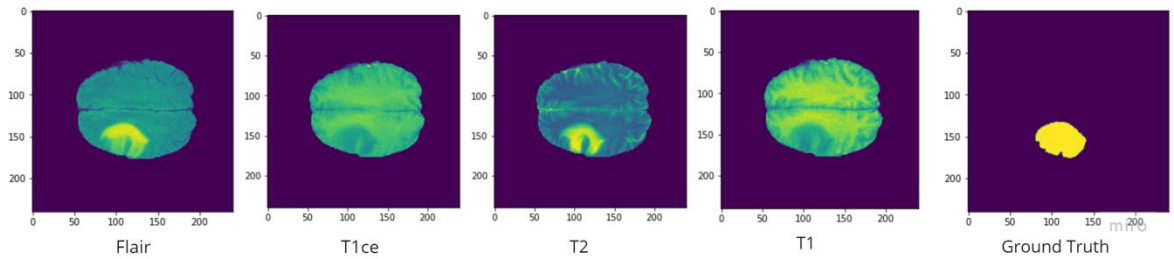

Figure 2 Brain Tumor on Each Modalities

On the other hand, the ground truth has four labels: label 0 (*background*), label 1 (*necrotic and non-enhancing tumor core / NET*), label 2 (*peritumoral edema / ED*), and label 4 (*enhancing tumor / ET*). The dataset is divided into three classes: *Whole Tumor*, which includes all tumor labels (1, 2, and 4); *Tumor Core*, which includes labels 1 and 4 but excludes label 2; and Enhanced Tumor, which contains just label 4. Figure 3 shows the visualization for each label based on the 75th slice of the first subject's image.

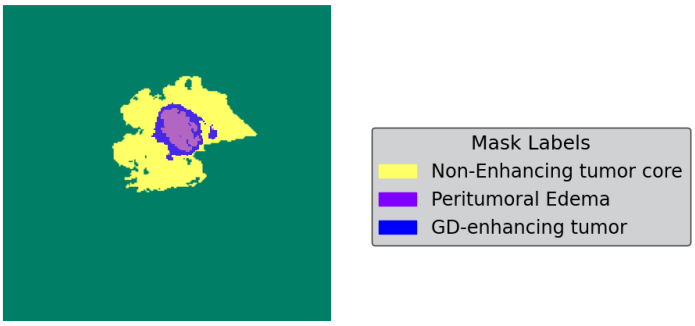

Figure 3 Ground truth with the labels information

2.3. Data Preparation, Preprocessing, and Augmentation

On this part will be explained about data preprocessing and augmentations is used on this research. Data preprocessing and augmentations is used to reduce the noise of the data and make the model more general to analyze brain tumor disease on the new data test because the lack of datasets size.

2.3.1. Data Preparation

In this part, the datasets are separated into two categories: data *training* and data *testing*, with 80% and 20% presentations respectively. Based on the whole dataset of 369 data subjects, 274 will be used for *training* and the remaining 95 for *testing*.

2.3.2. Preprocessing

Normalizations and *cropping* are utilized in this work as data pretreatment to improve model performance while reducing computation range number and targeted object detections.

a. Normalization

The goal of image normalization is to convert the array values of 3D brain tumor MRI pictures into a range of 0 to 1. The original distribution on the image as you can see on the Figure 4 Histogram Distribution Image Range Value, it shows the range value of image is between 0 until 625.

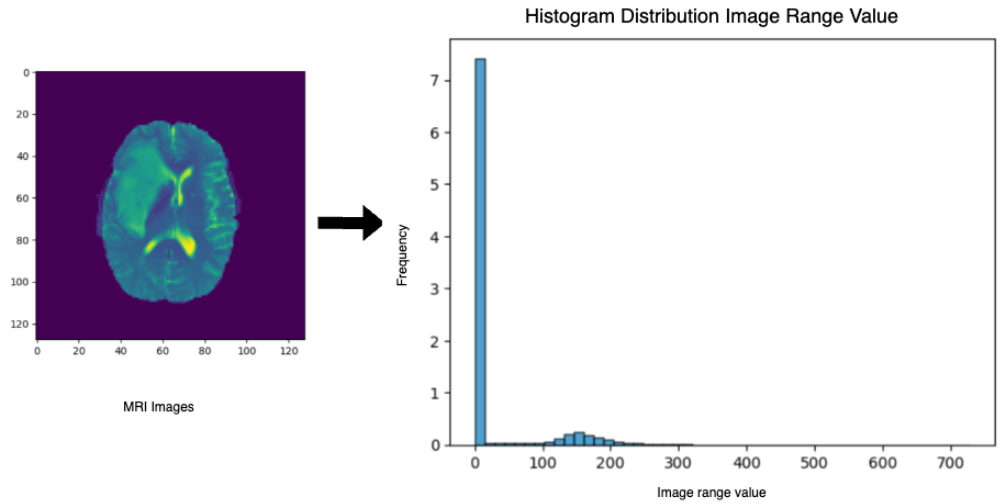

Figure 4 Histogram Distribution Image Range Value

This scaling speeds up the training process and enhances overall performance. The formula for these normalizations can be found on the equation (2.1). The normalizations resulted in a range of 0 to 1 on the value image, whereas the previous value ranged from 0 to 625.

$$Normalization = \frac{data - \min(data)}{\max(data) - \min(data)} \tag{2.1}$$

b. Cropping

On the other hand, Cropping is used to accelerate the training process and target the object tumor more precisely. To crop 3D brain tumor MRI images, resize them from 240×155×155 to 128×128×128. The cropping is centered on the image coordinates to highlight the brain tumor's target location. The representation of the cropping image, as seen on the Figure 5.

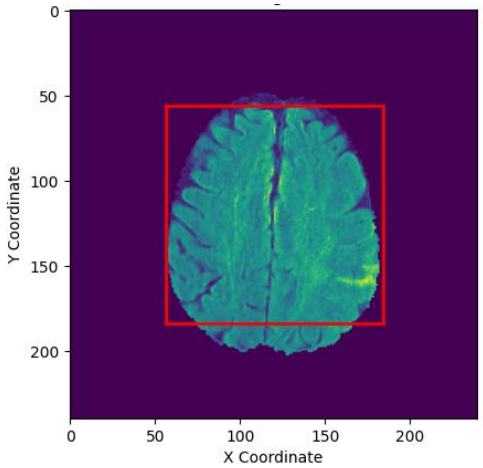

Figure 5 Brain Tumor Cropping Target

The image in the red area will serve as the input image for the training process. As a result, Figure 6 will show the differences between the images before and after the cropping operation. The cropped image appears closer than the pre-cropped image. This effect is because the information supplied by image cropping makes the image more relevant to the target object. This cropping is applied to both the input and ground-truth images.

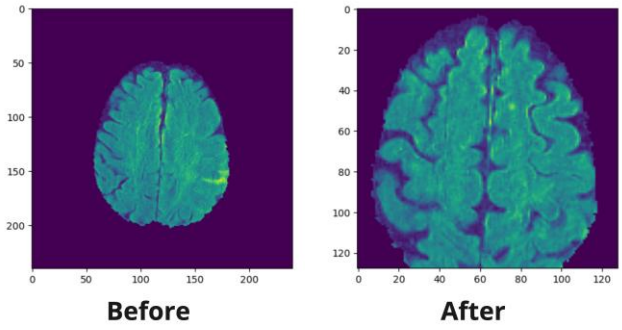

Figure 6 Before and After Cropping

2.3.3. Augmentations

Because of a shortage of entire data and to make the data more diverse, we used augmentations in this study, including *random gaussian noise* and *random flip*.

a. *Random Gaussian Noise* or *Gaussian White Noise*

Is a sort of random noise that follows a Gaussian (normal) distribution with mean (average) and set standard deviation. These settings can be changed to produce the appropriate noise level. Experimenting with Random Gaussian Noise seeks to assess the robustness of the created architecture approach and deep learning models when dealing with diverse data. The results of before and after augmentations can be see on the Figure 7.

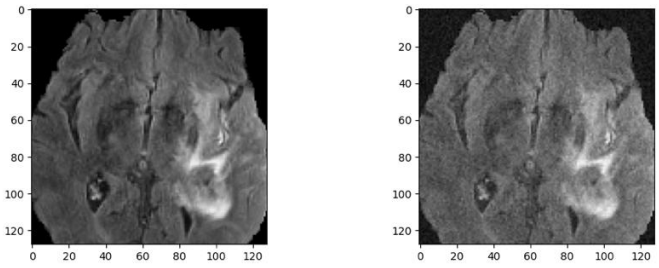

Figure 7 Before (Left) and After (Right) *Random Gaussian Noise*

b. *Random Flip*

This image flipping may be done in a variety of directions, including horizontal and vertical. In this investigation, random flips were applied with a probability of 0.5 along the X-axis (horizontal), Y-axis (vertical), and Z-axis (picture depth), allowing randomization to occur separately on each axis, with each axis having the same chance of being flipped.

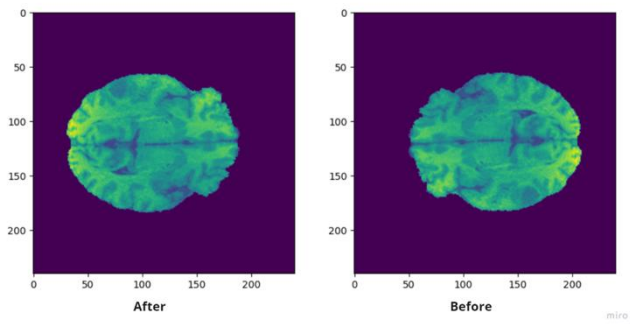

Figure 8 Illustration after and before *random flip*

2.4. Architecture 3D S-UNet

The 3D S-UNet is a combination of 3D ShuffleNet as an encoder and 3D U-Net as a decoder that produces brain tumor MRI segmentation images with little resource utilization.

2.4.1. Encoder 3D S-UNet

In the first layer, the researchers created a 3D convolution block with three components: *Conv3D*, *batch normalization*, and the *LeakyReLU* activation function. The formula for *LeakyRelu* can be seen on (2.2).

$$f(x) = \begin{cases} x, if\ (x \geq 0 \\ ax, if\ x < 0 \end{cases} \tag{2.2}$$

On that formula x work as the input value, and a as the small positive constant (e.g.,0.01) that controls the slope for negative x values. *LeakyReLU* was chosen as the activation function because it can overcome dying ReLU. This occurs when a neuron in a specific layer repeatedly generates zero output, indicating that it cannot contribute to the resultant model throughout the training process. The researchers employed three stages in the encoder part of the created 3D ShuffleNetV2, each with a different number of blocks: 4, 8, and 4 respectively. These steps act as repeating processes, allowing the resultant model to acquire more complicated representations as the network depth grows. The number of input channels rises with each stage, beginning with 32 in the first block, then 64, and finally 128 in the last block. Each repetition in these steps is built with an *Inverted Residual Block*, which involves channel shuffling. The entire architecture of 3D S-UNet can be depicted on Figure 9.

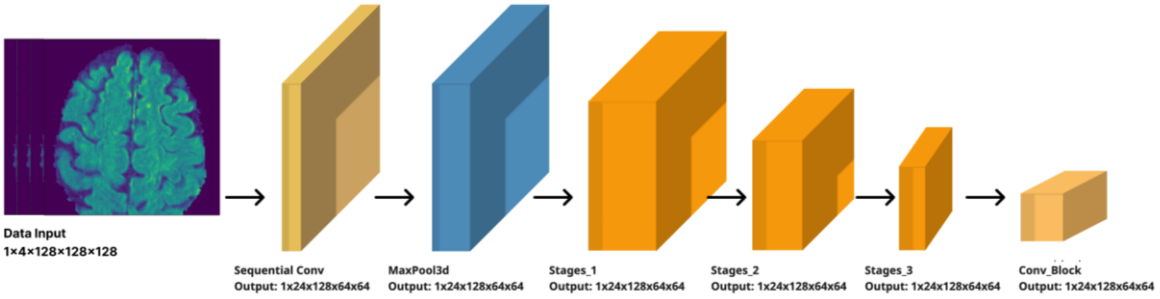

Figure 9 Architecture Encoder 3D S-UNet

2.4.2. Decoder 3D S-UNet

A convolution block, similar to the U-Net architecture, is added before the decoding process begins, consisting of *Conv3d* and *BatchNorm3d* functions. The researchers employed four decoder blocks to restore an image from 1×256×16×8×8 to its original size of 1×24×128×64×64. The number of 24 is the results of *Conv3D* process on encode architecture with the *kernel size* 3x3x3 and *stride* 1x2x2. Each decoder block consists of a *ConvTranspose3D* function for spatial data upsampling and a ConvBlock. The graphic Figure 10 shows an illustration of the decoder 3D S-UNet.

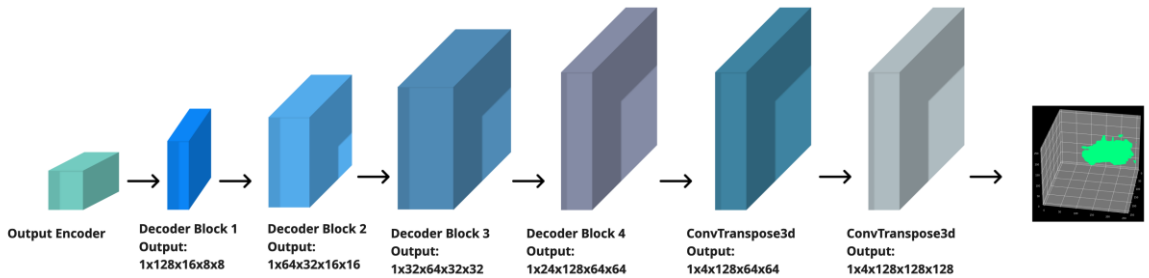

Figure 10 Architecture Decoder 3D S-UNet

2.5. Training 3D S-UNet

The modality order employed in this study was Flair, T1, T1ce, and T2. Only cropping without normalizing was used on the ground-truth photos to ensure that the stored labels remained identical. Ground-truth photos are cropped to 128×128×128, then converted to 3×128×128×128, with 3 representing the amount of brain tumor classes: total tumor, enhanced tumor, and tumor core. The augmentation methods used were Random Gaussian Noise and Random Flip. The training process in this work took place on a DGX-A100 system with 1/4 of an NVIDIA A100 GPU and 40 GB of GPU RAM, and the designed architecture was implemented using the PyTorch library.

2.6. Evaluation Metrics and Performance

Evaluation metrics are used to assess the quality of a machine learning model or its output statistics. The evaluation criteria employed in this study to examine the accuracy performance of the model generated by the developed architecture are dice, sensitivity, and specificity.

2.6.1. Dice Coefficient

The *Dice Coefficient Similarity*, or *Dice Coefficient*, is a statistical method for determining the similarity of two sets of data.

$$Dice = \frac{2TP}{FP + 2TP + FN} \tag{2.3}$$

2.6.2. Sensitivity

It is utilized to calculate the amount of true positives (TP) produced, allowing for the identification of the number of augmenting tumors that the Dice loss approach does not identify.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.4}$$

2.6.3. Specificity

It is one of the metrics used to assess a model's ability to prevent false positives (FPs).

$$Specificity = \frac{TN}{TN + FP} \tag{2.5}$$

2.6.4. Memory Consumption

In this study, we used PyTorch's built-in function, *PyTorch profiler*, to assess GPU memory use. This function calculates how much memory is allocated and reserved in the architecture. The unit used in this study is *Gigabytes* (GB).

2.6.5. Inference Time

In this study, inference time refers to how long it takes the architecture to train for one inference on a calculation utilizing a *CPU* and *CUDA*. The goal is to measure how long it takes for the built architecture to train. *FLOPS*, or *Floating Point Operations Per Second*, is a measure of computer performance that shows the number of floating-point operations a computer system can execute. It also influences inference speed measurement. It can alternatively be described as the number of floating-point multiplication-addition operations used in a model to assess latency, speed, and computational complexity.

3. Results and Discussion

3.1. Evaluation Metrics

These measures quantify the overlap and similarity between a model's projected segmentation and the actual ground truth. A higher score on these parameters suggests a better-performing architecture, implying that it is the best option.

Table 1. Results Evaluation Metrics

Metrics	Architecture		
	3D S-UNet	**3D Mobile U-Net**	**3D U-Net**
WT Dice	**83%**	68%	89%
WT Sensitivity	**85%**	72%	89%
WT Specificity	**88%**	82%	91%
TC Dice	**75%**	54%	80%
TC Sensitivity	**83%**	78%	81%
TC Specificity	**77%**	64%	82%
ET Dice	**65%**	55%	73%
ET Sensitivity	**67%**	70%	72%
ET Specificity	**74%**	67%	79%

Based on the Table 1. Architecture 3D S-UNet had the highest score on whole tumor (WT) segmentations which the overall score is up to 80%, that is also happen on tumor core (TC) which architecture 3D S-UNet still had the highest score than 3D Mobile U-Net. Finally, on the enchanced tumor (ET) 3D S-UNet only had the highest score on Dice and Specificity only, while sensitivity architecture 3D Mobile U-Net had the best score. Overall, 3D S-UNet received the best Dice score, indicating that it outperformed others in terms of spatial overlap and segmentation accuracy rather than 3D Mobile U-Net. In the other hand, when compared 3D S-UNet with 3D U-Net the 3D U-Net is still had the highest best accuracy. In practice, this means that the segmentation outputs are more dependable for downstream activities (such as diagnosis, treatment planning, or object analysis), particularly in applications that need precise region matching.

3.2. Performance

Table 2. Performance Metrics

Architecture	Performance Measure			
	Memory Consumption (GB)		**Inference Time**	
	Allocated	**Reserved**	**CPU (s)**	**CUDA (ms)**
3D S-UNet	**0.56**	**1.71**	**2.643s**	**135.881ms**
3D Mobile U-Net	0.58	2.43	3.673s	253.103ms
3D U-Net	1.76	4.39	4.047s	368.828ms

As shown in Table 2, the 3D S-UNet design had the lowest memory consumption, indicating that it used the fewest memory resources during the training phase. The same trend was observed during inference, where 3D S-UNet, despite its low memory utilization, outperformed Mobile U-Net by 1.03 seconds on CPU and 117.222 milliseconds on CUDA. In contrast, 3D S-UNet can minimize allocations, and memory consumption by up to two times compared to 3D U-Net and had the fastest inference time on CPU and CUDA up to 30% than 3D U-Net.

4. Conclusion

According to the evaluation results, the 3D S-UNet design performed the best across all criteria which compared with another lightweight deep learning. The shuffling mechanism contributes significantly

to ShuffleNet's accuracy. Although group convolutions significantly cut computational costs, they naturally limit information sharing between feature groups. By combining ShuffleNet with U-Net, the suggested 3D S-UNet is highly effective for medical picture segmentation, especially in resource-constrained contexts.

References

[1] S. Kedia *et al.*, "Brain Tumor Programs in Asia and Africa: Current Status, Challenges, and Future Perspectives," *World Neurosurg*, vol. 175, pp. e1041–e1048, 2023, doi: https://doi.org/10.1016/j.wneu.2023.04.067.

[2] U. Raghavendra *et al.*, "Brain tumor detection and screening using artificial intelligence techniques: Current trends and future perspectives," *Comput Biol Med*, vol. 163, p. 107063, 2023, doi: https://doi.org/10.1016/j.compbiomed.2023.107063.

[3] G. Tabatabai *et al.*, "Molecular diagnostics of gliomas: the clinical perspective.," *Acta Neuropathol*, vol. 120, no. 5, pp. 585–592, Nov. 2010, doi: 10.1007/s00401-010-0750-6.

[4] Y. T. Udaka and R. J. Packer, "Pediatric Brain Tumors," *Neurol Clin*, vol. 36, no. 3, pp. 533–556, 2018, doi: https://doi.org/10.1016/j.ncl.2018.04.009.

[5] M. Aljabri and M. AlGhamdi, "A review on the use of deep learning for medical images segmentation," *Neurocomputing*, vol. 506, pp. 311–335, 2022, doi: https://doi.org/10.1016/j.neucom.2022.07.070.

[6] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med Image Anal*, vol. 42, pp. 60–88, 2017, doi: https://doi.org/10.1016/j.media.2017.07.005.

[7] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges," *J Digit Imaging*, vol. 32, no. 4, pp. 582–596, 2019, doi: 10.1007/s10278-019-00227-x.

[8] K. Kamnitsas *et al.*, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med Image Anal*, vol. 36, pp. 61–78, 2017, doi: https://doi.org/10.1016/j.media.2016.10.004.

A. Beers *et al.*, "Sequential 3d u-nets for biologically-informed brain tumor segmentation," *arXiv preprint arXiv:1709.02967*, 2017.

[9] L. Dai, T. Li, H. Shu, L. Zhong, H. Shen, and H. Zhu, "Automatic Brain Tumor Segmentation with Domain Adaptation," in *BrainLes@MICCAI*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:59943040

A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," in *BrainLes@MICCAI*, 2018.

[10] H. Xu, H. Xie, Y. Liu, C. Cheng, C. Niu, and Y. Zhang, "Deep Cascaded Attention Network for Multi-task Brain Tumor Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:204028162

[11] M. Aly and A. S. Alotaibi, "EMU-Net: Automatic Brain Tumor Segmentation and Classification Using Efficient Modified U-Net," *Computers, Materials and Continua*, vol. 77, no. 1, pp. 557–582, 2023, doi: https://doi.org/10.32604/cmc.2023.042493.

[12] W. Zafar *et al.*, "Enhanced TumorNet: Leveraging YOLOv8s and U-net for superior brain tumor detection and segmentation utilizing MRI scans," *Results in Engineering*, vol. 24, p. 102994, 2024, doi: https://doi.org/10.1016/j.rineng.2024.102994.

[13] C.-H. Wang, K.-Y. Huang, Y. Yao, J.-C. Chen, H.-H. Shuai, and W.-H. Cheng, "Lightweight Deep Learning: An Overview," *IEEE Consumer Electronics Magazine*, pp. 1–12, 2022, doi: 10.1109/MCE.2022.3181759.

[14] S. Chattopadhyay, A. Dey, P. K. Singh, and R. Sarkar, "DRDA-Net: Dense residual dual-shuffle attention network for breast cancer classification using histopathological images," *Comput Biol Med*, vol. 145, p. 105437, 2022, doi: https://doi.org/10.1016/j.compbiomed.2022.105437.

[15] D. Das, D. R. Nayak, S. V Bhandary, and U. R. Acharya, "CDAM-Net: Channel shuffle dual attention based multi-scale CNN for efficient glaucoma detection using fundus images," *Eng Appl Artif Intell*, vol. 133, p. 108454, 2024, doi: https://doi.org/10.1016/j.engappai.2024.108454.

[16] N. Kim, S. Lee, J. Kim, S. Y. Choi, and S.-M. Park, "Shuffled ECA-Net for stress detection from multimodal wearable sensor data," *Comput Biol Med*, vol. 183, p. 109217, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.109217.

[17] S. Sangui, T. Iqbal, P. C. Chandra, S. K. Ghosh, and A. Ghosh, "3D MRI Segmentation using U-Net Architecture for the detection of Brain Tumor," *Procedia Comput Sci*, vol. 218, pp. 542–553, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.036.

[18] M. Pendse, V. Thangarasa, V. Chiley, R. Holmdahl, J. Hestness, and D. DeCoste, "Memory efficient 3d u-net with reversible mobile inverted bottlenecks for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2020, pp. 388–397.

[19] S. Bakas *et al.*, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge," *arXiv preprint arXiv:1811.02629*, 2018.

[20] S. Bakas *et al.*, "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features," *Sci Data*, vol. 4, no. 1, p. 170117, 2017, doi: 10.1038/sdata.2017.117