

Hyperparameter Optimization in UNet Variants for Improved Brain Tumor Segmentation

Eman Abdulaziz Ghani Aldhafer^{1,2*}, Amir Lakizadeh¹

¹ Department of Computer Engineering and Information Technology, University of Qom, Qom 37100, Iran

² Department of Computer Science, Faculty of Education for Girls, University of Kufa, Kufa 54001, Iraq

Corresponding Author Email: emana.alddaaher@uokufa.edu.iq

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310109>

ABSTRACT

Received: 23 July 2025

Revised: 21 October 2025

Accepted: 5 November 2025

Available online: 31 January 2026

Keywords:

brain tumor segmentation, UNet architecture, hyperparameter optimization, medical image segmentation, deep learning, magnetic resonance imaging

Accurate segmentation of brain tumors from magnetic resonance imaging (MRI) plays a crucial role in clinical diagnosis and treatment planning. In recent years, deep learning-based approaches, particularly convolutional neural networks (CNNs), have achieved remarkable success in medical image segmentation tasks. Among these models, the UNet architecture and its variants have demonstrated strong capability in capturing hierarchical image features. However, the performance of these networks is highly dependent on the selection of appropriate training hyperparameters. This study systematically investigates the influence of key hyperparameters on the segmentation performance of the original UNet model and several representative variants, including UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net. Specifically, the effects of learning rate (LR), batch size (BS), patience, and loss functions are analyzed using the Lower-Grade Glioma (LGG) Segmentation Dataset, a widely used benchmark dataset for brain tumor segmentation. Multiple experimental configurations are evaluated to identify the optimal parameter settings for each model. The networks are trained using the Adam optimizer and assessed using two widely adopted segmentation metrics: Dice Score and Jaccard Similarity. Experimental results demonstrate that hyperparameter selection significantly affects model convergence and segmentation accuracy. Among the tested configurations, a LR of 0.001, patience value of 6, and a BS of 32 consistently provide stable training and improved performance across most models. Furthermore, Dice Loss and DiceBCE Loss achieve better segmentation accuracy than Tversky Loss. Among the evaluated architectures, IRU-Net achieves the highest segmentation performance on the LGG Segmentation Dataset. These findings highlight the importance of systematic hyperparameter tuning for improving UNet-based brain tumor segmentation and provide practical guidance for optimizing deep learning models in medical image analysis.

1. INTRODUCTION

A brain tumor is a proliferation of abnormal cells or a mass of aberrant tissue in the brain's cortex that can be either benign (non-cancerous) or malignant (cancerous) [1-4]. Brain tumors are considered a serious medical illness and can cause mortality if not detected in the early stages [2-5]. Fast and efficient detection or prediction of brain tumors in the early stages of their occurrence can have a positive impact on patients' successful treatment outcomes [6, 7]. In the field of brain imaging, magnetic resonance imaging (MRI), is the gold standard technique for visualization of brain tumor [5, 8]. The MRI technique is used to locate tumors, assess their development, and monitor their responses to treatment [9].

The accurate segmentation of brain tumors that using medical data is essential for tumor classification since it removes the influence of non-tumor tissues [4, 10]. However, manual segmentation is a process that requires a lot of labor, time-consuming, and particularly complicated [11]. Deep learning (DL) networks have recently demonstrated superiority over traditional methods for autonomously

segmenting brain tumors. This success can be attributed to the advancements in the technology of graphics processing units (GPUs) and central processing units (CPUs), large datasets being available, and learning algorithm development. One of the DP algorithm types, convolutional neural networks (CNNs), have shown promising results in the detection of brain tumors using MRI and other medical images, especially after the proposal of the UNet model to the public in 2015 [2, 9, 10, 12, 13]. The UNet represents one of the essential DP networks for medical image segmentation, featuring a U-shaped architecture with skip connections that enables the accurate identification of the tumor regions in a brain MRI image [8, 14, 15]. The structure of the network comprises two components: a contracting path, referred to as the encoder, and an expansive path, known as the decoder, with a skip connection between them [9, 15]. Over the years, researchers have proposed many successful modifications and additions to the original UNet architecture to enhance the network's performance in medical image segmentation while preserving the U-shaped structure [16, 17]. The performance of the original UNet approach and its variants depends on the choice

and selection of the appropriate hyperparameter values, which is essential for achieving optimal segmentation results for the brain tumor task [2, 5].

The hyperparameters are parameters that the user sets prior to the training process to control the behavior of learning algorithms. They can be categorized into two types: the first one specifies the network structure, such as the number of filters, kernel size, and hidden layers; the second indicates the network training parameters, such as the number of epochs, batch size (BS), and learning rate (LR). The selection of parameters greatly influences the model's performance and final result, although learning algorithms do not automatically modify them [4, 9, 18].

The primary objective of this work is to investigate the effect of training hyperparameters on the performance of the original UNet model and its variants for brain tumor segmentation using the Lower-Grade Glioma (LGG) Segmentation Dataset. This study's contributions include the following:

1. Different model hyperparameters, including LR, patience, number of epochs, BS, and loss functions, are used to demonstrate how they affect the UNet model and its variants.

2. Several well-known UNet variants, including UNet3+ [19], the Multiscale Residual Dilated convolution neural network (MSRD-UNet) [20], Alex-UNet [15], and IRU-Net [21], were chosen due to their architectural variety and proven effectiveness in medical image segmentation tasks.

3. The chosen DL models (UNet, UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net) are employed, and a comparison among them is performed to show the effect of hyperparameters on their performance. For a clearer analysis, one hyperparameter was changed at a time, allowing easier observation of its effect on segmentation results for each model separately.

4. Jaccard Similarity and Dice Score, were utilized as evaluation metrics to assess the segmentation performance for each hyperparameter value of each model.

The structure of the manuscript is outlined as follows: Section 2 presents the related work. Subsequently, the details of the U-Net, its variants, and hyperparameters are discussed in Section 3. Section 4 delves into the experimental results, while Section 5 concludes the paper.

2. RELATED WORKS

Several articles have been conducted regarding the impact of hyperparameters on the accuracy of UNet models. They have tried to determine the most efficient network performances based on hyperparameters such as LR, BS, optimizer algorithms, etc. Pamungkas et al. [22] conducted a comprehensive analysis of the effect of hyperparameter tuning for brain tumor segmentation, trained with a ResNet-UNet on the LGG data. Experiments are performed using various ResNet backbones (ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152), and the effectiveness of them is compared based on different LRs (0.0001, 0.001 0.01), optimization algorithms (Adam, SGD, RMSProp), and sigmoid activation functions. The results indicated that using smaller LR, especially 0.0001, along with deeper models such as ResNet-50 and ResNet-152, led to better segmentation outcomes, achieving dice similarity coefficients (DSC) as high as 0.928 and Mean IoU of up to 0.9. Tapasvi et al. [2]

introduced a hybrid method for brain tumor identification and segmentation that integrates the U-Net architecture with the moth flame optimization (MFO) algorithm. The aim of the suggested model is to improve segmentation performance by optimizing hyperparameters, including LR and architectural depth, via MFO. The model outperforms Unet and Unet++ in MSE, PSNR, and Tversky by 65.16%, 28.87%, and 40.30%, respectively. The work of Zhu et al. [23] is focused on the process of segmenting brain tumors from multimodal MR images using edge-based and deep semantic features. The method has two main parts: a deep semantic segmentation network that operates on the UNet model that works with different types of MRI data and an edge detection module that makes the tumor edges more accurate. Cross-entropy loss and dice loss were used to train the model, helping the network find a good balance between accurately classifying individual pixels and overlapping regions. An evaluation of the proposed technique was carried out on the BraTS19 dataset, which consisted of 335 patient cases and four different types of tumors. Other models, like U-Net, Attention-Net, and 3D-Net, were compared to their model using Dice Score as the main benchmark. These results confirmed that the hybrid approach outperformed all others in terms of precision for small and irregularly shaped tumors. Also, the proposed method showed consistent performance regardless of model or hyperparameter configuration, suggesting that it could be used in many clinical datasets and different segmentation challenges. Fahsi et al. [8] investigated the impact of hyperparameter optimization on segmentation of brain tumors with the UNet architecture. In this case, an attempt was made to change the hyperparameter configuration of the basic UNet model, which includes the number of layers, dropout rates, LR, batches, and epochs, among others. The model was tested on the BRATS18 and BRATS20 datasets, and the results were measured using the Dice Coefficient metrics. The results demonstrate that the modified UNet architecture achieved higher DSC in core tumor segmentation compared to the original UNet. Jasmine et al. [10] proposed a deep network to segment brain tumor using You Look Only Once (YOLO) CNN with a focus on hyperparameter optimization to arrive at the optimal value. The model's performance was assessed using three distinct hyperparameters: anchor box number, batches, and learning method, employing the BRATS dataset for HGG and LGG segmentation as well as the REL dataset. Using four anchor boxes, 16 batches, and the Adam optimizer, the suggested methodology achieved effectiveness in segmentation. HGG segmentation has 0.9688 mean average precision and 0.9841 recall. LGG segmentation has a mean average precision of 0.8435 and recall of 0.8552. Zheng et al. [24] suggested an improved U-Net design to deal with the drawbacks in brain tumor segmentation, such as the difficulty of the traditional models' ability to accurately identify edge details and effectively reuse feature information. The method employed a hybrid dilated convolution (HDC) module integrated into a sequential encoder-decoder architecture to assist the network in capturing spatial features at multiple scales. Also, the authors proposed a custom loss function, which is designed to better segment complex and unclear regions of tumors that are usually challenging to diagnose. The experimental evaluation demonstrated that the enhanced model achieved superior performance when compared to traditional methods, reporting notable improvements in DSC, precision, and Hausdorff distance. These results highlight the potential of the proposed model to provide more accurate tumor delineation, ultimately

contributing to more efficient and reliable automated brain tumor diagnosis systems. The DL model for semantic segmentation of brain tumors based on a hierarchical residual attention network was proposed by Sun et al. [25]. The model was tested on two publicly available datasets and was shown to outperform several segmentation methods that are deemed to be state-of-the-art. The suggested system is based on the nested residual attention network (NRAN) architecture, which combines and residual networks and attention mechanisms. The NRAN is composed of several nested residual units, each containing a number of convolutional layers and attention blocks. These blocks are crucial as they allow the network to concentrate on the relevant features while filtering out extraneous information. Furthermore, to improve segmentation precision while retaining spatial information, the researchers added skip connections links the encoder and decoder parts of the network. The authors compared their

suggested method to a number of other basic segmentation techniques, including Attention UNet, UNet, VNet, in order to assess it. They used the Dice Score to measure the similarity of predicted and ground truth segments and evaluation of their work. Moreover, the authors performed sensitivity to analyze how segmentation performance was affected by varying hyperparameters and structural preferences.

3. METHODS

This section describes the architecture of UNet and its other variants, such as UNet3+, MSRD-UNet, Alex-UNet and IRU-Net. Also, this section provides some of the key training hyperparameters, including LR, patience, Adam optimizer algorithm, BS, epochs and loss functions.

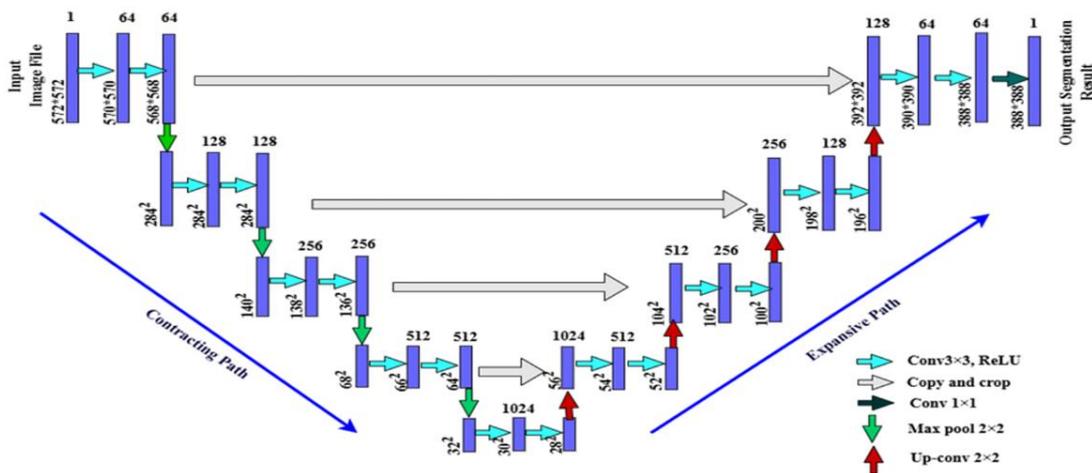


Figure 1. UNet structure

3.1 UNet architectures

3.1.1 UNet

UNet, one of the deep artificial networks originally designed for medical image segmentation tasks, takes its name from the "U"-shaped structure [22, 26]. It is featured by its unique encoder-decoder structure, which includes skip connections, which facilitates the accurate delineation of objects in images [27, 14]. UNet's architecture contains two symmetrical paths: the contracting encoder path and the expansive decoder path [28]. The encoder path is used to extract features from the input image, and it is made up of a series of convolutional layers and max pooling layers [8, 27, 29]. The decoder part consists of a sequence of up-sampling and deconvolutional layers to create a fully segmented image [2, 29]. UNet incorporates skip connections, which link the corresponding layers from the encoder and decoder paths, to enhance reconstruction performance. These connections are fundamental in improving the precision of segmentation by recovering the spatial information that has been down-sampled in the encoder, especially for thin and delicate structures in the image. [14, 28, 29]. UNet network structure is shown in Figure 1.

3.1.2 UNet3+

UNet 3+ is a full-scale connected UNet architecture with extensive deep supervision, which improves the segmentation map's spatial awareness and boundary detection while using

fewer parameters [17]. In UNet 3+, full-scale skip connections affect both the encoder and decoder layers and the intra-connections among decoder subnetworks [19]. In the UNet 3+ architecture, full-scale skip connections are employed to aggregate feature maps from multiple scales. This approach includes smaller and equal-sized feature maps from the encoder, as well as larger-scale maps from the decoder, all of which are fused and integrated into each decoder layer. This multiscale fusion significantly improves the model's capability in handling organ segmentation tasks across various spatial resolutions [16, 17, 19]. Additionally, deep supervision is incorporated by learning hierarchical feature representations from the aggregated full-scale feature maps. For implementation, the final output of each decoder sub-path undergoes a sequence of operations: it is first processed through a conventional 3×3 convolution layer, followed by bilinear up-sampling, and finally passed through a sigmoid activation function to produce the segmentation prediction [16, 17]. The UNet 3+ method outperforms UNet and UNet++, emphasizing organs and establishing consistent boundaries even for small objects.

3.1.3 MSRD-UNet

MSRD-UNet is an end-to-end model used a U-shaped structure and designed to improve medical image segmentation performance. It has three main improvements compared to the regular U-Net: a squeeze and excitation attention mechanism, updated skip connections, and

multiscale residual dilated blocks [20]. The approach replaced the traditional encoder and decoder with deeper blocks (MSRDB), which are made of two parallel dilated convolutions and a base convolution, which also acts as a residual link. This structural form can effectively combine multiscale features with multiscale gradient alleviation without adding parameters [20, 30]. Additionally, the redesign of the skip connection improves the spatial and semantic alignment between the two paths by merging feature maps from many encoder layers before transferring them to the decoder. Moreover, the SE modules, applied after the skip connections, assist in emphasizing informative features while suppressing irrelevant activations [20]. Experimental results and both quantitative and qualitative findings indicated that the MSDR-UNet model surpassed other U-Net-based approaches.

3.1.4 Alex-UNet

Alex-UNet is a light network that uses a pre-trained AlexNet in its encoder to improve the effectiveness of medical image segmentation, especially for skin lesions [31]. Alex-UNet takes advantage of AlexNet's architecture, which consists of eight layers with a large receptive field in the initial two layers [32]. This integration lets the model use transfer learning, which means it can extract features better, train faster, and generalize better. The encoder and decoder were built using the first six layers adopted from AlexNet. Each encoder layer comprises four components: convolution (Conv), rectified linear unit (ReLU), and max pooling (MP). Transposed convolutions (TConv) in each decoder layer concatenate the feature map with the encoder layer, doubling its size. Batch normalization, convolution, and rectified linear unit layers are implemented subsequent to this concatenation [15]. One of Alex-UNet's best features is its large receptive field in the very first layer. This is because the first AlexNet layer has a large kernel size (11×11), which is crucial for the classification and localization tasks required in semantic segmentation [31]. The Alex-UNet network is considered to be a lightweight network in comparison to the U-Net network since it significantly reduces the number of parameters, the amount of memory that is consumed, and the number of floating-point operations. Alex-UNet is much more effective than U-Net and other of its variants in terms of skin lesion segmentation [15].

3.1.5 IRU-Net

IRU-Net is an end-to-end DL network that combines the benefits of the GoogLeNet (Inception) with U-Net to enhance medical image segmentation performance, which includes the identification and segmentation of brain tumors [21, 33]. The IRU-Net structure consists of two main parts: an encoder part and a decoder part, each composed of six layers. The IRU-Net uses GoogleNet and residual connections as an encoder, allowing the model to gather various feature levels and combine them from different layers [34]. Each encoder layer consists of two Inception layers, 1×1 convolution, a rectified linear unit, and 2×2 spatial max pooling. After the encoder extracts the features, then they transmitted to the decoder part, which consists of an inception block with residual connections to enhance multi-scale feature representation and improve information flow [21]. The encoder layer is concatenated directly to the corresponding decoder layer via skip connections [35].

3.2 Hyperparameters setting

Hyperparameters are predetermined values established before training the DP model, such as UNet, and do not undergo any modifications during the training process [2, 12, 22]. They are essential in influencing model behavior and performance, training stability, and adaptation to new data. Therefore, an appropriate selection of hyperparameter values allows the best outcomes [12, 36]. The hyperparameter used in this paper is as follows:

3.2.1 Learning rate

LR is a hyperparameter that controls the amount of change made to model weights throughout each iteration [22, 37]. It determines the step size for each iteration towards the ideal solution. A high LR can lead to unstable convergence, which causes the model to overshoot optimal points during the optimization process. On the other side, a low LR may lengthen the training time or result in the model becoming stuck at a local minimum. Depending on the dataset and model architecture, the LR is often heuristically set between 0.01 and 0.0001, contingent upon the dataset and model complexity [22, 36].

3.2.2 Patience

Patience is the number of epochs without improvement. The value zero signifies training ends when performance degrades between epochs [8].

3.2.3 Batch size

BS is the crucial hyperparameter that refers to the number of training samples processed before the network weights are modified in a single iteration [18, 22, 36]. To accommodate the CPU hardware's memory requirements, the BSs are represented as powers of two. The selection of BS profoundly influences training stability and efficiency, with the optimal size typically ascertained through experimentation [9]. For instance, when using a small BS of 16 or 32, the model receives weight updates more frequently. The increased frequency can help it identify the best solutions faster, but it needs more iterations. In contrast, the model is able to utilize a greater amount of data before updating its weights with larger BSs, such as 128 or 256. This results in a more stable training process, but it also requires a greater amount of GPU memory. Determining the optimal BS typically necessitates experimentation, as an excessively large BS makes the model less generalizable, while an excessively small BS can make training noisy and unstable. To achieve a balance between speed and stability, empirical research proposes a BS ranging from 32 to 256 [22, 36, 37].

3.2.4 Epochs

Number of epochs (no-epochs) denotes the number of times the dataset has been employed for network training [18]. The impact of epoch choice on training is significant. A low number of epochs may result in the model's ineffective learning, causing underfitting. Conversely, a high number of epochs may result in the model memorizing the training data, leading to overfitting and subsequently diminished performance on new data. Methods like early stopping are commonly employed to mitigate overfitting. This method monitors the model's performance on a validation set and terminates training when no additional enhancement in performance is detected over multiple epochs. The no-epochs

is usually chosen between 10 and 100, contingent upon the dataset and model complexity [22, 36].

3.2.5 Loss function

The loss function is a key element of DL; it denotes the cost function that will be used by the model for error minimization. It facilitates the model's learning to enhance segmentation predictions throughout numerous training iterations, thereby minimizing false positives and false negatives [8, 22]. MRI segmentation tasks commonly use the following loss functions:

- Dice Loss is inspired by the DSC. The Dice loss directly optimizes the overlap between predicted and ground truth masks [22, 38]. It is represented as:

$$D_{Loss}(y, p) = 1 - \frac{2yp + 1}{y + p + 1} \quad (1)$$

where, the standard ground truth is denoted by y , while the predicted mask by the model is denoted by p .

- The Tversky loss function, derived from the Tversky index, generalizes the Dice loss to enhance the balance between precision and recall in the segmentation of highly imbalanced data using DL networks [39, 40]. Tversky loss is defined as:

$$T_{index} = \frac{TP}{(TP + \alpha FP + \beta FN)} \quad (2)$$

$$T_{Loss} = T_{index} - 1 \quad (3)$$

In this context, TP refers to true positives, FP refers to false positives, and FN refers to false negatives. The parameters α and β control the magnitude of penalties for FPs and FNs .

- DiceBCE Loss merges binary cross-entropy (BCE) loss with dice loss to make sure that the segmentation is precise at the pixel level and maintains a consistent overall shape [41]. The BCE formula is given by:

$$BCE_{Loss}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \quad (4)$$

where, y represented the actual value and p represented the predicted value. The $DiceBCE_{Loss}$ function represent as:

$$DiceBCE_{Loss} = D_{Loss} + BCE_{Loss} \quad (5)$$

3.2.6 Adam optimizer algorithm

The optimizer is an essential component of CNN training, which dictates how the model updates its weights according to gradient computations [42]. Adam is a first-order gradient optimization algorithm that is based on an adaptive lower-order moment prediction [43]. It maintains a different LR for each parameter, and these rates are automatically adjusted as the training process progresses. Adam demonstrates robust performance in deep networks, leading to its extensive application in medical image segmentation and classification [22].

4. EXPERIMENTAL RESULTS AND ANALYSIS

This section will describe the experimental environments and the dataset used to evaluate the UNet and its variants,

along with the evaluation metrics and implementation details. Finally, the results will be discussed.

4.1 Dataset

This research employed the LGG Segmentation Dataset, as obtained from Kaggle [44]. A fluid-attenuated inversion recovery (FLAIR) sequence is included in the pre-surgery imaging data, and there are anywhere between 20 and 88 MRI slices of the patient's brain. This collection contains around 3,929 brain MRI images together with the manual FLAIR segmentation results that go with them. Every image's ground truth was created by hand using a standard annotation procedure, and it was verified by a qualified neuroradiologist.

4.2 Evaluation metrics

Dice and Jaccard coefficients are the most commonly metrics employed to assess the performance of brain tumor segmentation systems [15, 37]. They are particularly useful in imbalanced datasets [22]. The Dice Score (DS) is used to quantify the similarity or overlap between the reference mask and the predicted segmentation mask [39]. The DC equation is given by:

$$DS = \frac{2TP}{2TP + FP + FN} \quad (6)$$

Jaccard Similarity (JS) measures the extent to which the predicted mask and the ground truth mask overlap [39]. JS is presented in the equation:

$$JS = \frac{TP}{TP + FP + FN} \quad (7)$$

A higher DS and JS score indicates better segmentation accuracy, while lower scores imply poorer segmentation ability.

4.3 Experimental environments

The experimental work was executed with the PyTorch framework [45] in Google's Collaboratory environment, a cloud-based platform designed for machine learning and DL tasks. For the evaluation, the training and testing sizes associated with the LGG dataset were 3,179 and 693, respectively. The resolution of the input images was scaled to 256×256 pixels.

4.4 Results and discussions

The goal of this research is to assess the efficiency of UNet, UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net within the scope of the LGG Segmentation Dataset. In order to identify the optimal combination of LR, number of epochs, and dice loss, certain hyperparameters are tuned in the UNet networks used for brain tumor segmentation. This assessment aims to enhance Dice and Jaccard coefficients.

Table 1 presents a summary of the performance metrics for the standard UNet and its variants (UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net) in brain tumor segmentation, specifically focusing on the LR parameter. The LRs that were used for this evaluation were 0.01 and 0.001 to assess the influence of learning speed on model convergence, where

0.001 is commonly used as a standard due to its balance between stability and efficiency. Each value of the LR was tested along with varying patience values (10, 6, 2) to examine the effect of small and large patience values on model training and convergence. A BS of 32 and a total of 30 epochs were

employed, with the Adam Optimizer algorithm applied. Dice loss utilized as a standard loss function for segmentation tasks due to its effectiveness in addressing imbalance issues within datasets. The results were assessed using DS and JS, which are essential metrics for representing segmentation performance.

Table 1. Performance of UNet models with different LR and patience values, BS = 32 and no-epochs = 30

Learning Rate, Patience	0.01, 10		0.01, 6		0.01, 2		0.001, 10		0.001, 6		0.001, 2	
	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)
Model	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)	(DS)	(JS)
UNet.	0.807056	0.784733	0.816286	0.789818	0.785061	0.756818	0.819039	0.795337	0.844063	0.820667	0.801944	0.778351
UNet3+.	0.811874	0.788688	0.765312	0.739286	0.745041	0.719126	0.845773	0.822368	0.841387	0.818872	0.812093	0.787134
MSRD-UNet.	0.873785	0.843717	0.817902	0.790496	0.862691	0.832656	0.87529	0.848356	0.880406	0.853126	0.867362	0.838226
Alex-UNet.	0.819177	0.792574	0.837923	0.812538	0.81286	0.788237	0.868375	0.840081	0.867992	0.838578	0.869974	0.839762
IRU-Net.	0.83982	0.816752	0.83789	0.814501	0.787523	0.760406	0.892194	0.863023	0.9098	0.881218	0.867141	0.839179

Note: DS = Dice Score; JS = Jaccard Similarity; LR = learning rate; BS = batch size.

Table 2. Performance of UNet models with different BS values, LR = 0.001, patience = 6, and no-epochs = 30

Batch Size	4		8		16		32		64	
	DS	JS	DS	JS	DS	JS	DS	JS	DS	JS
Model	DS	JS	DS	JS	DS	JS	DS	JS	DS	JS
UNet.	0.759466	0.734647	0.815914	0.790758	0.801028	0.777229	0.844063	0.820667	0.806612	0.78434
UNet3+.	0.787635	0.762755	0.777047	0.752039	0.837531	0.814581	0.841387	0.818872	0.786002	0.761284
MSRD-UNet.	0.804611	0.77981	0.838068	0.811009	0.882659	0.853246	0.880406	0.853126	0.862068	0.834055
Alex-UNet.	0.774439	0.750149	0.837306	0.813465	0.840346	0.809512	0.867992	0.838578	0.85819	0.827125
IRU-Net.	0.835894	0.810329	0.862635	0.834855	0.873309	0.848058	0.898967	0.870468	0.896203	0.867556

Note: DS = Dice Score; JS = Jaccard Similarity; BS =batch size; LR = learning rate.

Table 3. Performance of UNet models with different loss functions, BS = 32, LR = 0.001, patience = 6, and no-epochs = 30

Loss Function	DiceLoss		TverskyLoss		DiceBCELoss		
	Model	DS	JS	DS	JS	DS	JS
UNET.		0.844063	0.820667	0.78887	0.766045	0.821479	0.800447
UNET3+.		0.841387	0.818872	0.858288	0.832475	0.877446	0.848364
MSRD_UNET.		0.880406	0.853126	0.869004	0.839145	0.848364	0.848364
Alex-UNet.		0.867992	0.838578	0.864128	0.830019	0.875566	0.847332
IRU-Net.		0.898967	0.870468	0.887269	0.860279	0.90579	0.877935

Note: DS = Dice Score; JS = Jaccard Similarity; BS =batch size; LR = learning rate.

As shown in Table 1, a LR of 0.01 leads to lower segmentation accuracy, higher loss values and unstable convergence. This effect is illustrated by the models with patience values of 2, 6, and 10, which performed poorly with greater loss values and lower DS and JS. By reducing the LR to 0.001, resulted in improved segmentation accuracy, which in turn led to lower loss values and higher DS and JS, especially for a patience value of 6. The IRU-Net, MSRD-UNet, Alex-UNet, UNet, and UNet3+ all have DS values of 0.9098, 0.880406, 0.867992, 0.844063, and 0.841387, respectively, and JS values of 0.881218, 0.853126, 0.838578, 0.820667, and 0.818872. Figure 2 shows how the LR affects the dice loss values for UNet and its variations when using two different LRs and three patience levels. The chart shows that using a LR of 0.001 and a patience of 6 leads to lower dice loss values for all UNet models, meaning there is less difference between the model's predictions and the real values.

Table 2 summarizes the performance of the standard UNet and its variants (UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net) for brain tumor segmentation using the BS hyperparameter. The evaluation was carried out utilizing five different BSs (4, 8, 16, 32, 64), with an optimal LR of 0.001 and patience of 6, the number of epochs as 30, and using the Adam Optimizer algorithm. Dice loss utilized as a standard loss function for segmentation tasks due to it is well-suited to dealing with dataset imbalances. The results were assessed using DS and JS, which are essential metrics for representing

segmentation performance.

Table 2 demonstrates that, among the five BS values, a BS of 32 provides optimal training stability and efficiency. The BS of 32 yielded scores of 0.898967, 0.880406, 0.867992, 0.844063, and 0.841387 for the IRU-Net, MSRD-UNet, Alex-UNet, UNet, and UNet3+ models in terms of DS, whereas the scores for JS were 0.870468, 0.853126, 0.838578, 0.820667, and 0.818872, respectively. The BSs varied from 2 to 64, with the findings indicating that larger BSs generally yielded higher accuracy; yet, this trend was inconsistent, as using the BS of 64 resulted in diminished training outcomes. The BS value of 32 resulted in reduced loss values across all UNet models, with loss reaching its minimum, as illustrated in the loss curves in Figure 3. Based on the findings, BS is one of the most important hyperparameters influencing model performance. The selection of appropriate BSs is heavily influenced by the model architecture and dataset. For instance, the deeper models, IRU-Net and MSRD-UNet, achieved higher values in DS and JS, as well as the designer's experience. The BS values must strike a balance between speed and stability.

Table 3 summarizes the performance of standard UNet and its variants (UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net) for brain tumor segmentation using different common loss functions. The evaluation was carried out using three different loss functions (Dice Loss, Tversky Loss, and DiceBCE Loss) with fixed optimal parameters of a 0.001 LR and a 32 BS, which were found by experimentation. There were 30 epochs,

and the Adam Optimizer method was utilized. The outcome was assessed using DS and JS, which are essential metrics for evaluating segmentation performance.

As shown in Table 3, The results indicate that Dice Loss and DiceBCE Loss outperformed Tversky Loss, delivering higher DS and JS. In terms of DS, the DiceBCE achieved 0.90579, 0.875566, and 0.877446 in the IRU-Net, Alex-UNet, and UNet3+ models; in terms of JS, it achieved 0.877935, 0.847332, and 0.84836, respectively. Dice Loss yielded results of 0.880406 and 0.844063 in DS, and 0.853126 and 0.820667 in the Alex-UNet and UNet models. Figure 4 displays the curves of the loss functions (Dice Loss, DiceBCE Loss, and Tversky Loss), illustrating their impact on the UNet model and its variants. Based on the results, although Dice Loss is effective on its own for segmentation tasks, using it with BCE

Loss can result in more balanced and precise outcomes. For example, combining BCE Loss with Dice Loss (DiceBCE Loss) resulted in greatly increased performance in the IRU-Net model with a DS of 0.90579 and JS of 0.877935.

Table 4 presents a comparison between the UNet model and the different models discussed in this paper, focusing on the number of parameters and the computational complexity represented by floating-point operations (FLOPs). The IR-UNet model needs less memory and FLOPs than UNet, UNet3+, and MSDR-UNet. However, it needs more memory and FLOPs than Alex-UNet. On the other hand, IRU-Net has a lower number of parameters than both UNet and UNet3+, but it possesses a higher parameter than Alex-UNet and MSDR-UNet.

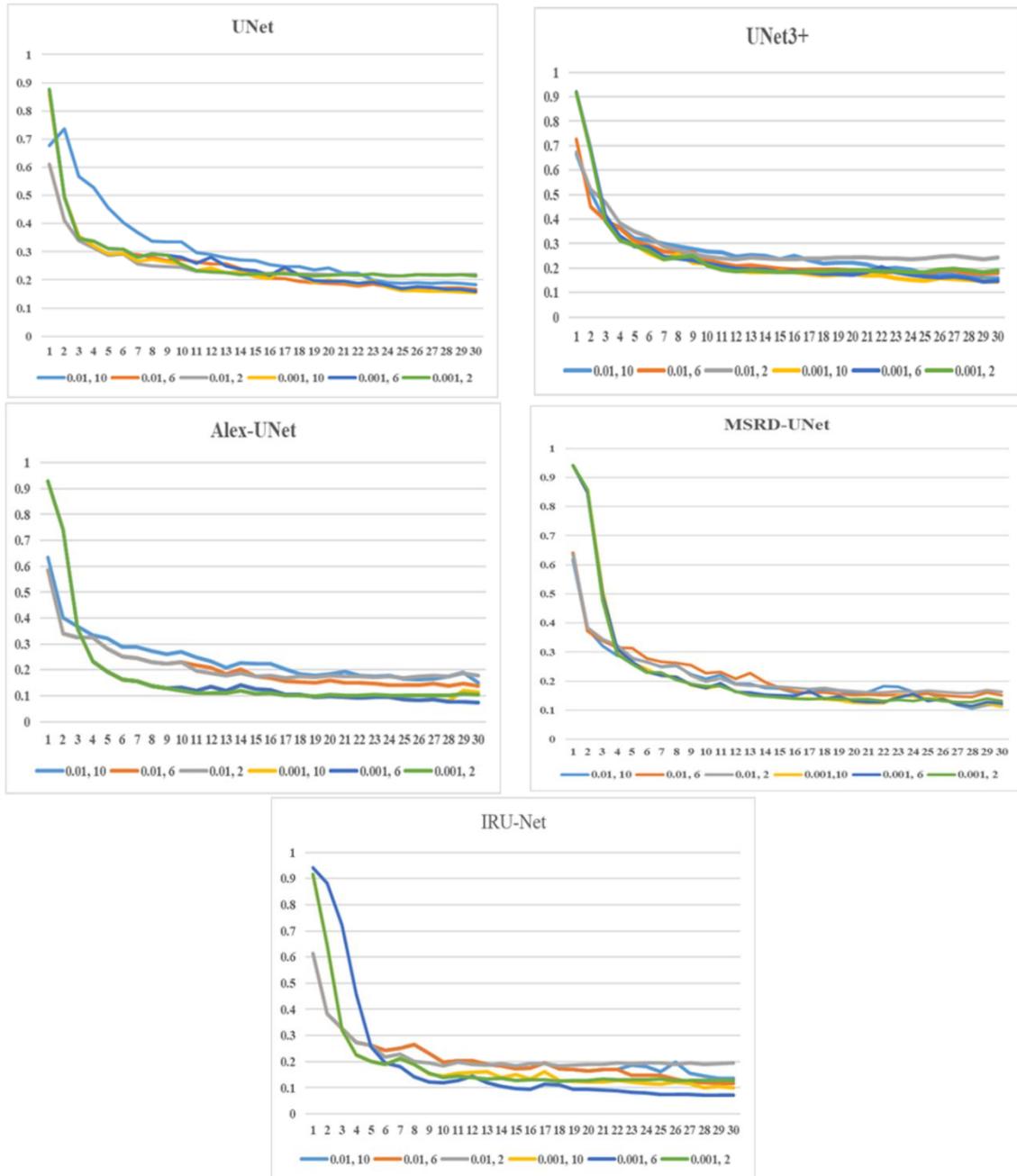


Figure 2. Dice loss curves for LGG Segmentation Dataset training on various UNet architectures using different LR and patience values

Note: LGG = Lower-Grade Glioma; LR = learning rate.

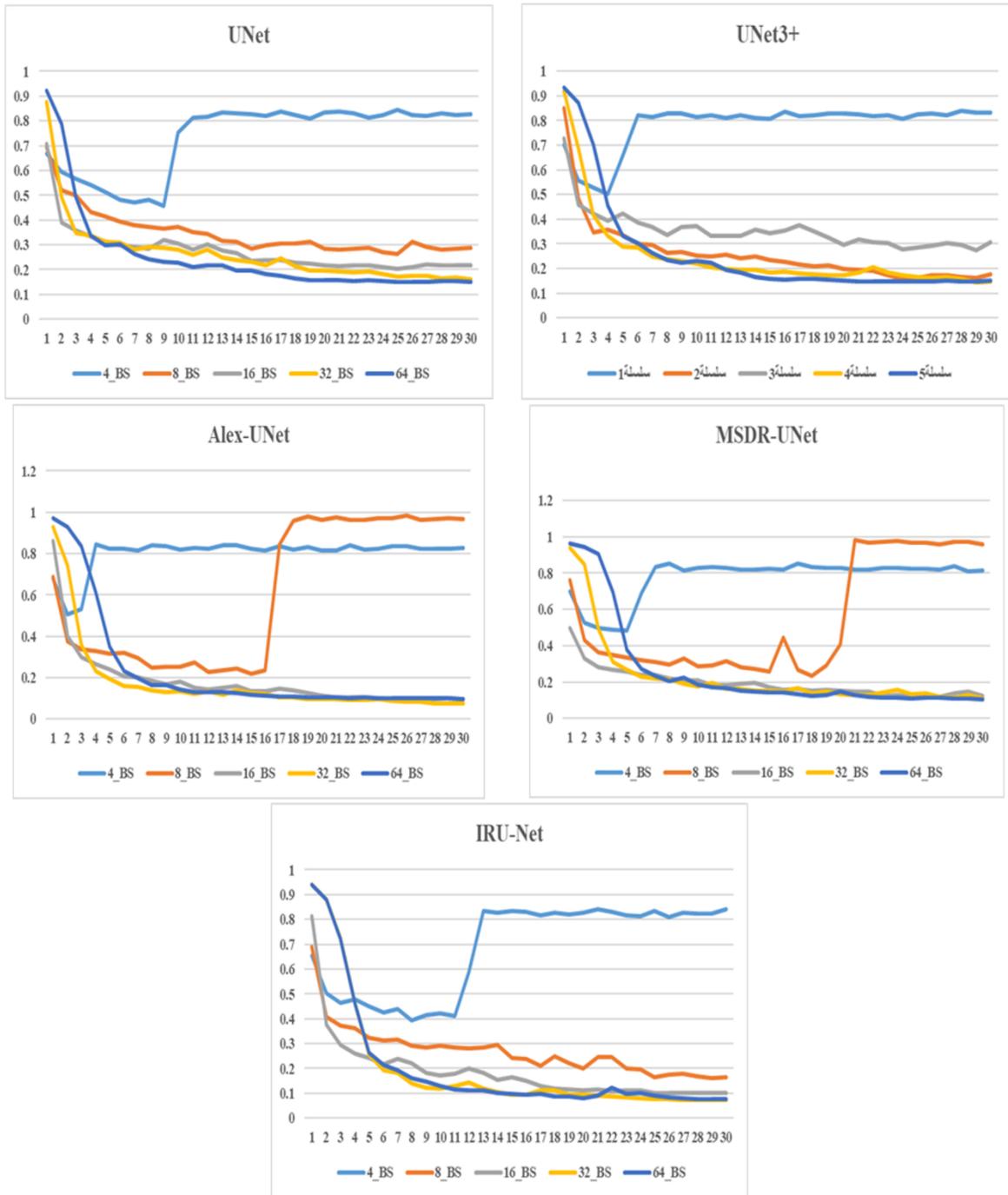
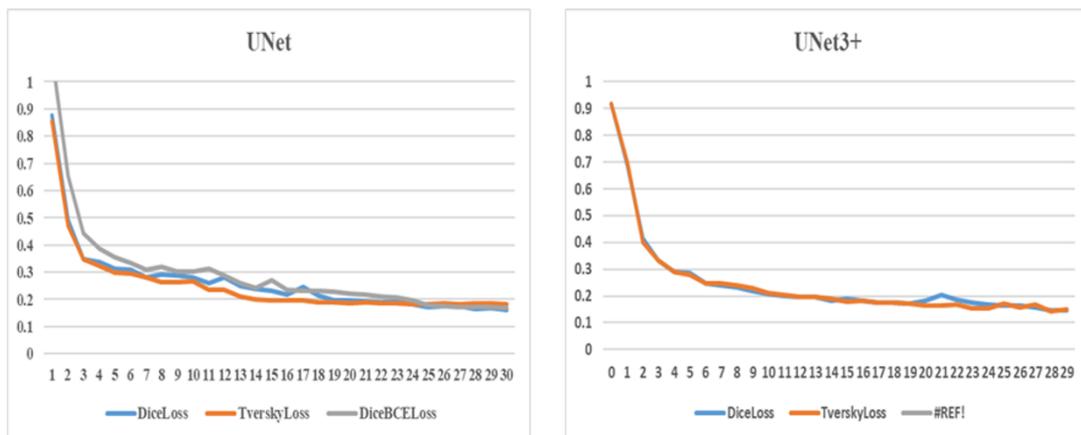


Figure 3. Dice loss curves for LGG Segmentation Dataset training on various UNet architectures using different BS values
 Note: LGG = LGG = Lower-Grade Glioma; BS = batch size.



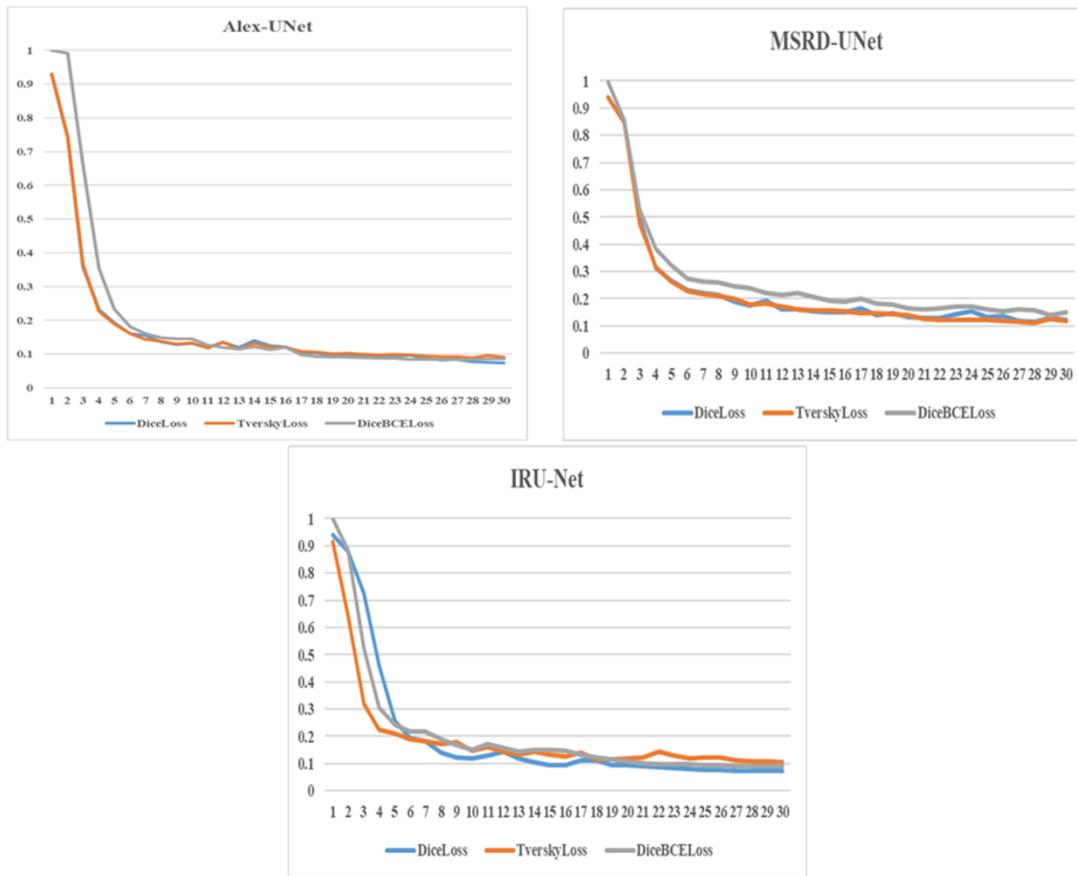


Figure 4. Dice loss curves for LGG Segmentation Dataset training on various UNet architectures using different loss functions
 Note: LGG = Lower-Grade Glioma; LR = learning rate.

Table 4. Resources utilized for UNet and its other variants

Model	UNET	UNet3+	MSRD-UNet	Alex-UNet
Total parameters	31,037,633	26,991,687	13,522,065	8,759,073
Forward/backward pass size for input size (MB): 0.75	1020 MB	2996.17 MB	1680.42 MB	161.46 MB
Parameters size (MB)	118.4 MB	102.97 MB	51.58 MB	33.41 MB
Number of FLOPS	48,327,294,976	198,733,561,866	35,548,439,552	2,711,067,936

5. CONCLUSIONS

This article presented the effect of hyperparameter tuning on the brain tumor segmentation performance of the basic UNet model and its four variants, including UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net, using the LGG brain MRI dataset. Different hyperparameters, including LR, patience, BS, and loss functions, which significantly impact segmentation, are tested to determine the best optimal values. The Adam optimizer algorithm was employed, as it is self-tuning and performs well in medical image segmentation and classification. The number of epochs used equals 35. The findings demonstrate that a LR of 0.001, in conjunction with a patience parameter of 6, improved segmentation accuracy across all models. The UNet, UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net models exhibited lower loss values, higher Dice Scores, and JS. In terms of BS, batches of 32 resulted in better performance than batches of 4, 8, 16, and 64. The BS equal to 32 achieved lower loss values and higher DS and JS in U-Net and its variants (UNet3+, MSRD-UNet, Alex-UNet, and IRU-Net). Additionally, the selection of loss functions influenced model performance, with Dice loss and DiceBDE loss outperforming Tversky loss when using fixed optimal

parameters of a 0.001 LR and a BS of 32. IRU-Net, Alex-UNet, and UNet3+ models with DiceBDE loss exhibited higher DS and JS. Meanwhile, Dice loss achieved higher values in DS and JS when applied to Alex-UNet and UNet models. The future studies aim to do more experiments by changing more hyperparameters, also implement other optimization techniques. More UNet architectures and loss functions will be studied, increasing the study's and models' reliability.

UNet3+, and MSRD-UNet. However, it needs more memory and FLOPs than Alex-UNet. On the other hand, IRU-Net has a lower number of parameters than both UNet and UNet3+, but it possesses a higher parameter than Alex-UNet and MSRD-UNet.

REFERENCES

- [1] Kumar, V.P., Pattanaik, S.R., Kumar, V.S. (2025). A heuristic strategy assisted deep learning models for brain tumor classification and abnormality segmentation. *Computational Intelligence*, 41(1): e70018. <https://doi.org/10.1111/coin.70018>

- [2] Tapasvi, B., Gnanamanoharan, E., Kumar, N.U. (2024). Brain tumor semantic segmentation using U-Net and moth flame optimization. *Journal of Intelligent Systems and Internet of Things*, 13(2): 334-346. <https://doi.org/https://doi.org/10.54216/JISIoT.130226>
- [3] Agarwala, S., Sharma, S., Shankar, B.U. (2022). A-UNet: Attention 3D UNet architecture for multiclass segmentation of Brain Tumor. In *2022 IEEE Region 10 Symposium (TENSYMP)*, pp. 1-5. <https://doi.org/10.1109/TENSYMP54529.2022.9864546>
- [4] Asiri, A.A., Shaf, A., Ali, T., Aamir, M., Irfan, M., Alqahtani, S. (2024). Enhancing brain tumor diagnosis: An optimized CNN hyperparameter model for improved accuracy and reliability. *PeerJ Computer Science*, 10: e1878. <https://doi.org/10.7717/peerj-cs.1878>
- [5] Ullah, N., Khan, M.S., Khan, J.A., Choi, A., Anwar, M.S. (2022). A robust end-to-end deep learning-based approach for effective and reliable BTd using MR images. *Sensors*, 22(19): 7575. <https://doi.org/10.3390/s2219757>
- [6] Rai, H.M., Chatterjee, K. (2021). 2D MRI image analysis and brain tumor detection using deep learning CNN model LeU-Net. *Multimedia Tools and Applications*, 80(28): 36111-36141. <https://doi.org/10.1007/s11042-021-11504-9>
- [7] Guo, X., Liu, T., Chi, Q. (2024). Brain tumor diagnosis in MRI scans images using Residual/Shuffle Network optimized by augmented falcon finch optimization. *Scientific Reports*, 14(1): 27911. <https://doi.org/10.1038/s41598-024-77523-2>
- [8] Fahsi, M., Nadir, M., Cheikh, M. (2022). A study of brain tumor medical image segmentation using U-Net. In *International Conference on Research in Applied Mathematics and Computer Science, Morocco*.
- [9] Magadza, T., Viriri, S. (2021). Deep learning for brain tumor segmentation: A survey of state-of-the-art. *Journal of Imaging*, 7(2): 19. <https://doi.org/10.3390/jimaging7020019>
- [10] Jasmine, R.A., Rani, P., Dhas, J.A. (2022). Hyper parameters optimization for effective brain tumor segmentation with YOLO deep learning. *Journal of Pharmaceutical Negative Results*, 13.
- [11] Núñez-Martín, R., Cervera, R.C., Pulla, M.P. (2017). Gastrointestinal stromal tumour and second tumours: A literature review. *Medicina Clínica (English Edition)*, 149(8): 345-350. <https://doi.org/10.1016/j.medcle.2017.06.045>
- [12] Román, K., Llumiquinga, J., Chancay, S., Morocho-Cayamcela, M.E. (2023). Hyperparameter tuning in a dual channel U-Net for medical image segmentation. In *Conference on Information and Communication Technologies of Ecuador*, pp. 337-352. https://doi.org/10.1007/978-3-031-45438-7_23
- [13] Rayed, M.E., Islam, S.S., Niha, S.I., Jim, J.R., Kabir, M.M., Mridha, M.F. (2024). Deep learning for medical image segmentation: State-of-the-art advancements and challenges. *Informatics in Medicine Unlocked*, 47: 101504. <https://doi.org/10.1016/j.imu.2024.101504>
- [14] Liu, L.L., Cheng, J.H., Quan, Q., Wu, F.X., Wang, Y.P., Wang, J.X. (2020). A survey on U-shaped networks in medical image segmentations. *Neurocomputing*, 409: 244-258. <https://doi.org/10.1016/j.neucom.2020.05.070>
- [15] Khalaf, M., Dhannoon, B.N. (2022). Skin lesion segmentation based on U-shaped network. *Karbala International Journal of Modern Science*, 8(3): 493-502. <https://doi.org/10.33640/2405-609X.3248>
- [16] Halder, A., Sau, A., Majumder, S., Kaplun, D., Sarkar, R. (2025). An experimental study of U-Net variants on liver segmentation from CT scans. *Journal of Intelligent Systems*, 34(1): 20240185. <https://doi.org/10.1515/jisys-2024-0185>
- [17] Jiangtao, W., Ruhaiyem, N.I.R., Panpan, F. (2025). A comprehensive review of U-Net and its variants: Advances and applications in medical image segmentation. *IET Image Processing*, 19(1): e70019. <https://doi.org/10.1049/ipr2.70019>
- [18] Lateef, R., Abbas, A.R. (2022). Tuning the hyperparameters of the 1D CNN model to improve the performance of human activity recognition. *Engineering and Technology Journal*, 40(4): 547-554. <http://doi.org/10.30684/etj.v40i4.2054>
- [19] Huang, H., Lin, L., Tong, R., Hu, H., et al. (2020). UNet 3+: A full-scale connected UNet for medical image segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 1055-1059. <https://doi.org/10.1109/ICASSP40776.2020.9053405>
- [20] Khalaf, M., Dhannoon, B.N. (2022). MSRD-UNet: Multiscale residual dilated U-Net for medical image segmentation. *Baghdad Science Journal*, 19(6): 48. <https://doi.org/10.21123/bsj.2022.7559>
- [21] Aldhafer, E.A.G., Lakizadeh, A. (2025). IRU-Net: Brain tumor detection in MRI images using the end-to-end inception residual UNet model. *International Journal of Advances in Soft Computing & Its Applications*, 17(2): 84-99. <https://doi.org/10.15849/IJASCA.250730.05>
- [22] Pamungkas, Y., Triandini, E., Yunanto, W., Thwe, Y. (2025). Impact of hyperparameter tuning on ResNet-UNet models for enhanced brain tumor segmentation in MRI scans. *International Journal of Robotics & Control Systems*, 5(2): 917-936. <https://doi.org/10.31763/ijrcs.v5i2.1802>
- [23] Zhu, Z., He, X., Qi, G., Li, Y., Cong, B., Liu, Y. (2023). Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal MRI. *Information Fusion*, 91: 376-387. <https://doi.org/10.1016/j.inffus.2022.10.022>
- [24] Zheng, P., Zhu, X., Guo, W. (2022). Brain tumour segmentation based on an improved U-Net. *BMC Medical Imaging*, 22(1): 199. <https://doi.org/10.1186/s12880-022-00931-1>
- [25] Sun, J., Li, J., Liu, L. (2021). Semantic segmentation of brain tumor with nested residual attention networks. *Multimedia Tools and Applications*, 80(26): 34203-34220. <https://doi.org/10.1007/s11042-020-09840-3>
- [26] Karthik, M., Thangavel, K., Sasirekha, K. (2024). Hyperparameter tuning of U-Net to breast pectoral muscle segmentation and classification using pre-trained models. *International Journal of Intelligent Systems and Applications in Engineering*, 12(20s): 372-382. <https://ijisae.org/index.php/IJISAE/article/view/5149>
- [27] Ehab, W., Huang, L., Li, Y. (2024). UNet and variants for medical image segmentation. *International Journal of Network Dynamics and Intelligence*, 3(2): 100009. <https://doi.org/10.53941/ijndi.2024.100009>
- [28] Huang, L., Miron, A., Hone, K., Li, Y. (2024). Segmenting medical images: From UNet to res-UNet and NNUNet. In *2024 IEEE 37th International Symposium*

- on Computer-Based Medical Systems (CBMS), Guadalajara, Mexico, pp. 483-489. <https://doi.org/10.1109/CBMS61543.2024.00086>
- [29] Gupta, A., Dhar, J. (2023). A hybrid approach for improving U-Net variants in medical image segmentation. arXiv preprint arXiv:2307.16462. <https://doi.org/10.48550/arXiv.2307.16462>
- [30] Alwan, A.H., Ali, S.A., Hashim, A.T. (2024). Medical image segmentation using enhanced residual U-Net architecture. *Mathematical Modelling of Engineering Problems*, 11(2): 507-516. <https://doi.org/10.18280/mmep.110223>
- [31] Bairagi, V.K., Gumaste, P.P., Rajput, S.H., Chethan, K.S. (2023). Automatic brain tumor detection using CNN transfer learning approach. *Medical & Biological Engineering & Computing*, 61(7): 1821-1836. <https://doi.org/10.1007/s11517-023-02820-3>
- [32] Kuang, Z.D. (2022). Transfer learning in brain tumor detection: From AlexNet to Hyb-DCNN-ResNet. *Highlights in Science, Engineering and Technology*, 4: 313-324.
- [33] Alsaif, H., Guesmi, R., Alshammari, B.M., Hamrouni, T., Guesmi, T., Alzamil, A., Belguesmi, L. (2022). A novel data augmentation-based brain tumor detection using convolutional neural network. *Applied Sciences*, 12(8): 3773. <https://doi.org/10.3390/app12083773>
- [34] Chen, S.R., Zhao, S.J., Lan, Q. (2022). Residual block based nested U-type architecture for multi-modal brain tumor image segmentation. *Frontiers in Neuroscience*, 16: 832824. <https://doi.org/10.3389/fnins.2022.832824>
- [35] Zhang, Q., Geng, G., Zhou, P., Liu, Q., Wang, Y., Li, K. (2024). Link aggregation for skip connection–mamba: Remote sensing image segmentation network based on link aggregation mamba. *Remote Sensing*, 16(19): 3622. <https://doi.org/10.3390/rs16193622>
- [36] Kumar, J., Dalal, N., Sethi, M. (2023). Hyperparameters in deep learning: A comprehensive review. *International Journal of Intelligent Systems and Applications in Engineering*, 12(4): 4015-4023.
- [37] Smith, S.L., Kindermans, P.J., Ying, C., Le, Q.V. (2017). Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489. <https://doi.org/10.48550/arXiv.1711.00489>
- [38] Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *International Workshop on Deep Learning in Medical Image Analysis*, pp. 240-248. https://doi.org/10.1007/978-3-319-67558-9_28
- [39] Salehi, S.S.M., Erdogmus, D., Gholipour, A. (2017). Tversky loss function for image segmentation using 3D fully convolutional deep networks. In *International Workshop on Machine Learning in Medical Imaging*, pp. 379-387. https://doi.org/10.1007/978-3-319-67389-9_44
- [40] Abraham, N., Khan, N.M. (2019). A novel focal Tversky loss function with improved attention U-Net for lesion segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy, pp. 683-687. <https://doi.org/10.1109/ISBI.2019.8759329>
- [41] Wazir, S., Fraz, M.M. (2022). HistoSeg: Quick attention with multi-loss function for multi-structure segmentation in digital histology images. In *2022 12th International Conference on Pattern Recognition Systems (ICPRS)*, Saint-Etienne, France, pp. 1-7. <https://doi.org/10.1109/ICPRS54038.2022.9854067>
- [42] Zaheer, R., Shaziya, H. (2019). A study of the optimization algorithms in deep learning. In *2019 Third International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, pp. 536-539. <https://doi.org/10.1109/ICISC44355.2019.9036442>
- [43] Stevens, E., Antiga, L., Viehmann, T. (2020). *Deep Learning with PyTorch: Build, Train, and Tune Neural Networks Using Python Tools*. Manning.
- [44] Buda, M. (2019). Brain MRI segmentation. <https://www.kaggle.com/datasets/mateuszbeda/lgg-mri-segmentation>.
- [45] Paszke, A., Gross, S., Massa, F., Lerer, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, vol. 32. https://proceedings.neurips.cc/paper_files/paper/2019/file_e/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.