Automatic segmentation of glioblastoma multiform brain tumor in MRI images: Using Deeplabv3+ with pre-trained Resnet18 weights

Fereshteh Khodadadi Shoushtari a, Sedigheh Sina a, b, Azimeh N.V. Dehkordi c, *

a Nuclear Engineering Department, Shiraz University, Shiraz, Iran
b Radiation Research Center, Shiraz University, Shiraz, Iran
c Department of Physics, Najafabad Branch, Islamic Azad University, Najafabad, Iran

ARTICLE INFO

Keywords:
- Enhanced tumor volume
- Semantic segmentation
- Deep learning
- Glioblastoma
- Deeplabv3+

ABSTRACT

Purpose: To assess the effectiveness of deep learning algorithms in automated segmentation of magnetic resonance brain images for determining the enhanced tumor, the peritumoral edema, the necrotic/ non-enhancing tumor, and Normal tissue volumes.

Methods and Materials: A new deep neural network algorithm, Deep-Net, was developed for semantic segmentation of the glioblastoma tumors in MR images, using the Deeplabv3+ architecture, and the pre-trained Resnet18 initial weights. The MR image Dataset used for training the network was taken from the BraTS 2020 training set, with the ground truth labels for different tumor subregions manually drawn by a group of expert neuroradiologists. In this work, two multi-modal MRI scans, i.e., T1ce and FLAIR of 293 patients with high-grade glioma (HGG), were used for deep network training (Deep-Net). The performance of the network was assessed for different hyper-parameters, to obtain the optimum set of parameters. The similarity scores were used for the evaluation of the optimized network.

Results: According to the results of this study, epoch #37 is the optimum epoch giving the best global accuracy (97.53%), and loss function (0.14). The Deep-Net sensitivity in the delineation of the enhanced tumor is more than 90%.

Conclusions: The results indicate that the Deep-Net was able to segment GBM tumors with high accuracy.

Introduction

One of the challenging issues in medical practice is the determination of the tumor location and its extension, i.e., tumor segmentation. Tumor segmentation is an essential step in evaluating the tumor situation and its treatment response follow-up [1]. Currently, the most common approach in the segmentation of the tumors is manual tumor delineation, which is a time-consuming process, which needs the long work of a group of clinical expert neuro-radiologists. Therefore, it can be easily affected by human error. Nowadays, the need for accurate and automatic diagnosis and treatment methods in health care is increasing, because of the increasing number of patients due to the population growth, and the rising demand for better, and faster health services. Recently, computer-aided techniques [2–10] have increased the accuracy and speed of the tumor segmentation process with the potential of reproducibility in medical applications and reducing the human dependency on the process. There are many challenging factors in the segmentation of complicated tumors like Glioblastoma Multiform, GBM, such as localization, accurate border delineation, and precise determination of tumor extension area [11,12]. GBM are usually diffused tumors with ambiguous boundaries and very different shapes and sizes. Magnetic Resonance Imaging (MRI) is an effective imaging technique to analyze brain GBM tumors. Different sequences of MRI can help to distinguish the different regions of the tumors [13–15]. The enhancing growth, and the rising demand for better, and faster health services. Recently, computer-aided techniques [2–10] have increased the accuracy and speed of the tumor segmentation process with the potential of reproducibility in medical applications and reducing the human dependency on the process. There are many challenging factors in the segmentation of complicated tumors like Glioblastoma Multiform, GBM, such as localization, accurate border delineation, and precise determination of tumor extension area [11,12]. GBM are usually diffused tumors with ambiguous boundaries and very different shapes and sizes. Magnetic Resonance Imaging (MRI) is an effective imaging technique to analyze brain GBM tumors. Different sequences of MRI can help to distinguish the different regions of the tumors [13–15].
tumor region is more clearly displayed in T1-weighted or in contrast-enhanced T1-weighted (T1ce) MR images [13]. Whereas, the edema and necrotic area are better highlighted in the Fluid Attenuation Inversion Recovery (FLAIR) images [13]. Therefore, the combination of the different MRI sequences in automatic brain tumor segmentation can be very useful in providing complementary information about the whole tumor and improving the segmentation performance.

Deep neural networks are capable of automatically extracting the appropriate features from the image and then learning and recognizing the pattern of the tumors and normal tissues, so the deep-learning algorithms have been widely used for tumor segmentation in the last few years [16–18]. Convolutional Neural Network (CNN) structure is the most popular structure among the deep neural networks for segmentation tasks [2,4,11,18,19]. So far, the different dimensional CNN with 2D [4,20], 2.5D [21], or 3D structures, [22–25] were used in tumor segmentation. 2D CNNs (slice-by-slice form) are extensively used in designing the deep-learning-based algorithm due to their less memory requirement. Using higher-dimensional CNN helps to preserve more image information at cost of more memory and time required. 2.5D CNN is used as a trade-off between inter-slice features and memory consumption [21].

The feature extraction capability of the network has a vital role in the CNN outcome. In deep CNNs, the convolutional layers are responsible for the automatic extraction of the features. Various structure designs [8,20,26,27] and different feature extraction techniques such as the combination of the local and global feature extraction [28], using a secondary training stage that trains the network by patches uniformly from the image [4] and fusing the 2D images in axial, sagittal, and coronal views to make 3D features and provide more information for the network [29–31] or different weight balance functions [32,33] were employed by the previous researches to enhance the feature extraction capability of the CNNs. A group of investigators [2,20,28,34] tried to optimize the previously reported architectures or developed structures with less computing need to make the process faster or to prevent the network from overfitting.

Pre-trained deep neural networks are also widely used in previous investigations for tumor segmentation. Reasonable results reported in the published works encouraged the investigators to develop their deep-learning segmentation algorithm based on the pre-trained networks [7,8,22,35–38]. The Deeplabv3+, introduced by Chen et al [39] is an encoder-decoder architecture and is especially well adapted to the prediction of pixel labeling. In this work, a deep learning-based algorithm (Deep-Net) was developed and well optimized for semantic segmentation of the glioblastoma tumor in MR images, using the Deeplabv3+ architecture, and the pre-trained Resnet18 initial weights. To improve the segmentation performance of the tumor sub-regions, the combination of two MRI modalities including T1ce and FLAIR images were employed to train the deep network and segment the whole brain images into four classes, i.e., The Gd-enhancing tumor, the peritumoral edema, the necrotic and non-enhancing tumor (NCR/ECD), and Normal tissue. Since, deep learning models are strongly dependent on the model hyper-parameters, such as Batch Size, Initial Learning Rate, Learn Rate Drop Factor, Epoch number, L2 Regularization, etc.; finding the optimized values for these parameters in such a high-dimensional space is a serious challenge, and can significantly affect the behavior of the training process, the performance of the trained model, and also the cost function value. This paper addresses the need for obtaining an optimized package of the model hyper-parameters, so far lacking in the scientific literature. Therefore, this work has focused on this issue by executing a lot of models with a wide investigating range for each hyper-parameter.

Materials and methods

Datasets

The Dataset used in the experiment was taken from the BraTS2020 training set [34,40,41]. The BraTS2020 dataset included sample multi-institutional routine clinically-acquired pre-operative multimodal MRI scans (T1-weighted, T1(ce)-weighted, T2-weighted and FLAIR) of 369 patients—293 glioblastoma/high-grade gliomas (GBM/HGG) and 76 low-grade gliomas (LGG), with accompanying ground truth labels for different tumor sub-regions manually drawn by a group of expert neuro-radiologists. The ground truth, GT, annotations comprise the GD-enhancing tumor (ET), the peri-tumoral edema (PE), and the necrotic and non-enhancing tumor core (NCR/NET). All 3D-MR images of the BraTS’2020 dataset have the dimension of 240 × 240 with 155 slices through the whole brain and skull stripped. In this work, two multimodal MRI scans, i.e., T1ce and FLAIR of 293 patients with high-grade glioma (HGG), were used for deep learning network training (Deep-Net).

Data preparation and image preprocessing

Since the developed Deep-Net used two-dimensional (2D) CNNs, first, all 3D data were converted into 2D slices from the axial view, and 90,830 2D slices of two MRI modalities were prepared. The slices with tumor information less than 0.5% of the whole brain volume were removed from the dataset to reduce the memory requirement and to improve the network performance [5]. Thus, 35,008 slices were picked up to form the dataset of this study. Secondly, labeled images were created according to the ground truth labels which were recognized by the group of radiologists to identify the different parts and distribution of the glioblastoma tumor. All regions out of the tumor and inside the brain were considered normal tissues. Therefore, as it can be seen in Fig. 1, each labeled image consists of five classes, i.e., normal tissue (blue), peritumoral edema (yellow), NCR/ECD (gray), the Gd-enhancing tumor (red), and background (black).

A few image processing steps were applied to the images before recruiting them in the training of the developed deep network. As the intensity value of the MR image depends on the imaging protocol and scanner used, the intensities in the original MR images were normalized to a value between 0 and 255. Tissue inhomogeneity may lead to significant misclassification in the segmentation procedure [42,43], therefore, A multiplicative intrinsic component optimization (MICO) [42], was also applied to the images to correct the inhomogeneity through a specific tissue and prevent the overlap between the ranges of the intensities of different tissues. In the next step, to limit the image to brain borders, the extra background parts of the original images with the dimension of 240 × 240 were cropped and then reduced to 224 × 224 using the nearest-neighbor interpolation technique, because the input of our developed deep neural network, Deep-Net, needs an image input with the size of 224 × 224 × 3, where 3 is RGB channels. Since the used data is grayscale, each image was repeated three times to fill the three channels. Fig. 1 illustrates the preparation and preprocessing steps applied to the images before using them as input images of Deep-Net.

Computing platform

This study was conducted on the MATLAB 2021a platform [44] using the Deep Learning Toolbox. The GPU, NVIDIA GeForce RTX 3080 with 12 GB RAM, in addition to CPU Intel Core i7-10700 K (@3.5 GHz) with 32 GB RAM were used on the Windows 10 system.

2–3-The Deep-Net architecture

The developed Deep-Net in this study is a pixel-wise segmentation network based on Deeplabv3+ architecture and the pre-trained Resnet18 initial weights [39]. The Deeplabv3+ is based on an encoder-decoder architecture as shown in Fig. 2.

Encoder: A pre-trained CNN extracts basic information from the input image for segmentation tasks [45]. The Encoder component is composed of convolution layers with a max-pooling layer and dilated convolutional layer, based on Atrous convolutions, the Atrous Spatial Pyramid Pooling (ASPP) module. The baseline CNN consists of 4
convolution layers with Rectified Linear Unit (ReLU) activation function and a kernel size of $7 \times 7$ in the first layer and $3 \times 3$ in others. After each convolution layer, there is a batch normalization layer (BN). After the first convolution layer, a $3 \times 3$ max-pooling layer was employed that shrinks the size of the feature map. The size reduction (downsampling factor) is controlled by the pooling size and the value, which corresponds to the horizontal and vertical increments at the positioning of pooling sub-windows. The produced feature map imports the ASPP module with rates 6, 12, and 18 (for three parallel $3 \times 3$ convolution branches with a default downsample rate of 16) to extract the multi-scale contextual information of images by creating a larger field of view [37]. This pathway leads to extracting the image context and informative features map from the input images but in reduced size and higher depth. The output stride value in an encoder part can be very effective in the accuracy and computation speed of the network, two possible stride values (8 and 16) are available in the Deeplabv3+ structure.

Decoder: The information extracted from the encoding step is used to create an output image similar to the size of the input image. The decoder block includes bilinear Up-sampling layers (deconvolution layers with stride 4), concatenation (concatenation layer takes inputs and concatenates them along a specified dimension), the SoftMax layer, and the classification layers. The decoder components recover the size of the feature map to the size of the input image and add the spatial resolution of the features to the map, which was lost during the downsampling. Since, low-level features have more spatial information and inclusion with high-level features enhances the segmentation performance, to preserve the intermediate low-level features produced in encoder layers, a $1 \times 1$ convolution is performed on the feature maps produced in the intermediate output of the encoder, and then this contextual information is concatenated with the up-sampled encoded features by an $8 \times 8$ convolution with stride 4. A $3 \times 3$ convolution filter is also applied and then an up-sampling process by a factor of 4 was employed again to produce the final segmentation output [39]. Lastly, a SoftMax non-linearity layer was employed that normalizes and assigns a multi-nominal distribution over the different labels of classes, and then, the pixels of each image were classified into 5 classes.

This structure needs a feature extraction network in its encoder part. A few of pre-trained networks (such as Resnet18 [46], Resnet50 [46], Resnet101 [46], mobilenetv2 [47], Xception [48] and Efficientnet [49]) are capable to employ as extractor at encoder part of Deeplabv3+. Among the possible networks, the Resnet18 weights as its initial weights were recruited in our algorithm. A combination of Deeplabv3+ and Resnet18 weights were adapted and optimized according to our data. Resnet18 is an 18-layer network and has trained on more than one million images from the ImageNet database [46]. This pre-trained network needs for calculation of 11.7 million parameters to be performed. The input images of Resnet18 are $224 \times 224 \times 3$. To get the pre-trained Resnet18, first, Deep Learning Toolbox™ Model for the Resnet18 network in MATLAB2021a software was installed. Then utilizing the “deeplabv3plusLayers” command, the layers and weights of Resnet18 were added to the Deeplabv3+ structure.

Training of the developed Deep-Net

Among the prepared 2D slices for all 293 patients, the 26256 slices of 220 patients were used as the training set, 1750 slices of 15 patients were employed as a validation set, and the 7002 slices of the remained 58 patients were recruited as a testing set. Four main classes were considered to evaluate the Deep-Net efficiency: enhanced tumor (ET), NCR/ECT, peritumoral edema (PE), and normal tissue (NT). One of the main challenges during the training process is the volume of each class or the imbalance between the number of pixels. This may cause a defect in the segmentation, because training is dominated by the classes with more pixels, like non-tumor pixels, and normal classes, while the algorithm goal focuses on the classes with the fewer pixels, like...
ET. To overcome this issue, different balance techniques such as median frequency balancing technique [50], inverse frequency balancing technique [51], and uniform class balancing technique [52] were employed to find the best balancing approach to adjust the class volume, and improve the efficacy of the Deep-Net for the smaller classes.

In this study, the cross-entropy loss function [53] was employed to guide the learning process to calculate the error between the Deep-Net output and the labeled ground truth, and the stochastic gradient descent (SGD) algorithm was utilized as an optimizer to update the weights and reach the minimum lost.

Deep-Net hyper-parameters optimization

Adjustment of the network hyper-parameters can significantly affect the network performance. Each hyper-parameter can directly change the weight of network layers and also affect the high features learning capability, which leads to a different convergence point [54]. Therefore, the following network parameters were investigated to find their optimal values and maximum Deep-Net performance.

Batch size (BS)

Since batch size controls the number of training samples in each iteration, it can have a great effect on the accuracy of the trained network. The advantage of using small batch size values is its less memory requirement, and fast training process. However, increasing the possibility of the noise effect, and also network divergence, is the disadvantage of small BS [55]. Very large batch sizes are usually accompanied by a drop in the generalization ability of the network [55]. A good investigation range of the batch size in deep neural networks is 8 to 128. Therefore, the batch sizes of 8, 16, 32, 64, and 128 were used in this study to find the optimum value.

Initial learning rate (ILR)

The ILR determines the learning speed of the training model and also affects the accuracy of the trained Deep-Net. A few different values in the range of 0.001 to 0.02 were checked and the ILR corresponding with the best performance without sacrificing the speed of the training was chosen as the optimum value. As a too high ILR may cause to lose of the local minima of the lost function, we limited our trial-and-error range of ILR to less than 0.02, besides the very low ILR may either take a long time to converge or get tramp in undesirable local minima, so the investigated range was considered more than 0.001.

Learn rate drop factor (LRDF)

The LRDF introduces the updated learning rate in each step by multiplying the initial learning rate by a determined drop factor. This parameter can be tuned to a value between 0 and 1. A few various drop factors in the range of 0.1 to 0.8 were assessed. The learning rate drop periods of 5 and 10 epochs were investigated in this study.
Epoch

The epoch number determines the number of passes of the entire dataset to the network. Most deep neural networks usually reach a relatively saturated accuracy after a specific epoch, after which the accuracy improvement is not significant. The optimal epoch was obtained as the epoch located at the 10% of the plateau region of the validation accuracy curve.

The shuffle function was also employed to riffle the data for each epoch to have different data for each batch. The data shuffling can help to decrease variance and keep the model general, and lessen the over-fit.

L2 regularization (L2R)

L2R was used to reduce the network complexity and avoid network overfitting. A few common values such as 0.0001, 0.005, and 0.01 were investigated to reach the best loss function for our developed Deep-Net. Of course, in this study, an extensive investigation for finding the optimal regularization rate was not performed.

Deep-Net evaluation with performance metrics

To quantify the performance of the trained Deep-Net, 20% of all the dataset was selected as test data, and a few metrics including global and mean accuracy, mean and weighted Intersection over Union, IoU, and Boundary F1 Score, BFscore, were calculated to assess the global performance of the Deep-Net in segmentation. The Sørensen-Dice Similarity Coefficient, DSC [56], and Intersection Over Union (IoU), also known as the Jaccard similarity coefficient (JSC), were computed to evaluate the similarity percentage between the Deep-Net predicted labels with the ground truth labels for each 5 considered classes. The tumor sub-regions were also grouped into three classes and the DSC and JSC were computed to assess the three constructed regions: enhanced tumor (ET), Tumor Core or TC (including the Enhanced tumor region with NCR \ECT), and Whole Tumor or WT (including the Tumor core and Edema region).

The performance evaluation metrics [57] used in this study are as follows:

1. Accuracy or global accuracy = \( \frac{TP + TN}{TP + TN + FP + FN} \)
2. Sensitivity, Recall, or true positive rate TPR = \( \frac{TP}{TP + FN} \)
3. Specificity, selectivity, or true negative rate, TNR = \( \frac{TN}{TN + FP} \)
4. Precision = \( \frac{TP}{TP + FP} \)
5. BF score = \( 2 \times \text{precision} \times \text{recall} / (\text{recall} + \text{precision}) \)
6. Dice Similarity Coefficient (DSC) = \( \frac{2TP}{2TP + FN + FP} \)

Fig. 3. A: correct classification fraction (CCF), or global accuracy of the trained Deep-Net at different epoch numbers. B: calculated loss function in validation process of the Deep-net.
\[ \text{(7) Intersection over Union, or JSC} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} = \frac{\text{(P} \cap \text{GT})}{\text{P} \cup \text{GT}} \]

Where TP is True-Positive, TN is True-Negative, FP is False-Positive, FN is False-negative, \( \text{P}_{\text{red}} \) is the predicted map, and the GT is the ground truth.

**Results**

**Deep-Net optimization results**

Fig. 3A shows the correct classification fraction, CCF, or the global accuracy of the optimal Deep-Net at different epochs respectively. The CCF increases by the epoch number until the curve reaches a plateau, and does not increase any further at higher epochs. The epoch number corresponding to 10% of the plateau region of the graph was taken as the stopping or optimum epoch. As shown in the figure, the 37th epoch is chosen as the optimal epoch, or the termination epoch for training the Deep-Net. The validation accuracy of the optimized Deep-Net at optimum epoch was achieved at 97.5%. In the other words, there is a 2.5% chance of failure for the trained Deep-Net to provide a correct classification when it is applied. Fig. 3B shows the variation of the cross-entropy loss function as a function of the epochs number. The loss function at the optimal epoch is 0.14. As shown in the figures, the improvement of the accuracy and loss values in greater epochs are not significant.

The developed Deep-Net was optimized for a few hyper-parameters. Figs. 4A to 4D illustrate the variation of the loss function and CCF with different values of the aforementioned hyper-parameters. Fig. 4A indicates that the small batch sizes are insufficient to train the network and improve the global accuracy. However, a further increase in the batch size from 64, gives rise to performance reduction and the batch size investigation was end up at 128. Based on the results shown in the Figure, the BS of 64 with the best network accuracy and loss was picked up as the optimum batch size. Fig. 4B displays the dependence of the accuracy and loss function on the ILR. As shown in the figure, the optimal ILR with minimum loss and maximum accuracy was obtained at ILR = 0.009. LRD factor was also inspected for a few values in two different drop periods. A reduction factor in the range of 0.1 to 0.8 was applied after every 10 epochs and the correct classification factor of the Deep-Net was calculated at each factor. The solid curve in Figure-4C represents the Deep-Net global accuracy for tested LRD with a drop period of 10 and the dashed curve illustrates the Deep-Net loss function values versus various LRDs at this drop period. As can be seen in the figure, the performance outperformed when the LRD was equal to 0.4. To further investigate the Deep-Net performance, the optimized value of the LRD, 0.4, was once tested on the network with a drop period of 5 and the results were compared with the drop period of 10. According to the results, the network, with a drop period of 5 gives an accuracy of 96.98%, and a loss of 0.165, which is not better than the drop period of 10 (accuracy = 97.5%, and loss = 0.14). Therefore, the combination of LRD = 0.4, and a drop period = 10, was more adequate and selected as the optimal set of hyper-parameters. Fig. 4D shows the evaluation of Deep-Net performance against different regularization rates. As shown in the figure, the regularization rate of 0.005 among the three evaluated regularization rates corresponds to better performance and was employed in the training process of Deep-Net. The results also implied that the L2 regularization rate of 0.0001 has signs of network overfitting.

The algorithm was also tested for two down sampling rates of 16 and 8. The down sampling rate significantly affects the processing time, a smaller down sample, results in a larger feature maps dimension (28 × 28), and a longer up-sampling process to recover the size of the image. The results indicate that different down sampling rates can influence the divergence process and consequently the algorithm efficacy. The results showed that the algorithm with a down sample rate of 8 reached a better global accuracy (97.5%) than the down sample rate of 16 (97.13%) with a cost of the slower training process (4 h and 30 min in return of 3 h and 30 min). This accuracy improvement impresses a 2–3% increment in the prediction of the whole tumor region and also a 2% increment on the enhanced tumor area. As the sampling rate becomes smaller, the number of weights that are applied to the valid feature region becomes larger, which makes the computing time of the algorithm to be longer. Using the down sample rate of 8 also makes the ASPP module rates to be double (i.e., rates of 12, 24, and 36). To speed up the training process (∼1.30 times faster) and reduce the memory consumption, a higher down sampling rate was used and tested. Down sample of 16 reduces the
training times, however, this may come at the cost of missed small objects and reduced accuracy at semantic boundaries. The calculated evaluation scores for a trained network with a down sample of 16 (validation accuracy 97.13%, validation loss 0.146) are as follows:

- Global accuracy (or CCF) calculated in confusion matrix for 5 classes: ET = 89.04%, PE = 95.7%, NCR\ECT = 87.37%, NT = 96.8% and BG = 99.55%
- DSC, Mean IoU (JSC), precision, sensitivity and BF score are 80%, 73.87%, 79%, 95% and 69.21%, respectively for PE class, 65.75%, 73.47%, 83%, 87% and 79% respectively for NCR\ECT class, 77.41%, 76.51%, 84%, 89% and 84.42% respectively for ET class and 98.18%, 96.52%, 99%, 97% and 88.41% respectively for NT class.
- DSC and JSC are 85.50%, 84.56%, 77.41% and 76.61%, 78.06%, 67.34% respectively for WT, TC and ET classes. The results of the trained network under down sample rate of 16 are also promising on different evaluation metrics with a higher speed in the training process.

**Performance metrics results of the optimized Deep-Net in class segmentation**

After choosing the best hyper-parameters, the trained Deep-Net was evaluated by 7002 test data. The mean overall performance of the developed semantic segmentation algorithm was evaluated with different performance metrics. The global and mean values of accuracy, mean IoU, weighted IoU, and mean BF score were found to be 97.5%, 94%, 85%, 95%, and 87% respectively.

Table 1 tabulates the confusion matrix of algorithm performance for 5 classes: peri-tumoral edema (PE) as class 1, NCR\ECT as class 2, Enhanced tumor (ET) as class 3, Normal tissue (NT) as class 4, and background (BG) as class 5. The confusion matrix of the Deep-Net gives a lot of information about the network performance. The confusion matrix revealed that the constructed Deep-Net shows high performance in classifying the five aforementioned classes correctly. The Deep-Net efficacy for all classes is above 87%. As shown in the Table 1, the diagonal elements of the CCF confusion matrix (sensitivity) for the five output classes were found to be 95.43%, 87.35%, 90.53%, 97.35%, and 99.44% for Classes 1 to 5 respectively. The sensitivity, specificity, and Type I error of Deep-Net can be easily calculated from the confusion matrix. The specificity was calculated to be 93.1%, 92.98%, 88.45%, 96.45%, and 99.11% for classes 1 to 5 respectively, and Type I error, false positive, were found to be 6.9%, 7.02%, 11.55%, 3.55%, and 0.09% for the five classes. However, the non-uniform distribution of the pixels into five different classes may cause the segmentation network to become biased towards the classes with larger volumes and increase the Type I error in smaller classes, but the problem was relatively solved using the median frequency balancing technique.

Then, to better analyze the segmentation power of the Deep-Net to classify the brain pixels into different classes, different scores (DSC or F1 score, mean BF score, Mean IoU or JSC, precision and sensitivity or recall) were used for network performance evaluation for the four main classes, which are listed in Table 2. The scores were not reported for the 5th class, or BG, due to its less importance. All these evaluation scores were measured with the same test data. The normal tissue (NT) has a larger volume between all classes and the trained Deep-Net for segmentation of the normal pixels just encounters a challenge in the pixels around the tumor or the pixels in the boundary of the brain. Therefore, specificity and type I error are very small for this class and as can be seen in Table 2, all calculated scores for this class are above the 97%. This table implies that the tumor region and its structure can be reliably segmented by the trained Deep-Net. DSC and JSC indicate the similarity of the segmented maps by Deep-Net with ground-truth maps. DSC above 67% was achieved for all tumor regions. An important class to be accurately segmented is enhanced tumor (ET). The attained sensitivity for ET is 90%. The Dice similarity coefficient for ET is 78%, which is among the highest obtained DSC score in the literature [2,3,20]. The high sensitivity (high accuracy) and high precision (high reliably reproducible) for all classes reveal that the trained Deep-Net is accurate and reliable enough to be used in clinical segmentation practice. The BF score measures how close the predicted boundary of an object matches the ground truth boundary and is defined as the harmonic mean (F1-score) of the precision and sensitivity values with a distance error tolerance (the default value is 0.75% of the length of the image diagonal). The value was calculated for each class and tabulated in Table 2.

To compare the results with the results of previous investigations, the tumor region was grouped into three classes and all evaluation scores were measured for the new three classes too: Class A = Whole Tumor (WT), Class B = Tumor Core (TC), Class C = Enhancing Tumor (ET). The classification of the pixels into the three mentioned classes was performed on the predicted segmentation label maps and ground truth label maps, and a 0 and 1 mask was prepared for each class. Then, the two prepared maps were compared. The DSC of 87%, 85%, and 78% and JSC of 79%, 79%, and 68% for Class A, Class B, and Class C reveal that the segmented maps by the Deep-Net are strongly in agreement with the ground truth maps.

**Visualize evaluation of the Deep-Net performance**

The eight exemplary segmentation result of the developed and trained algorithm with the respective ground truth images is included in Fig. 5. Four sample slices were chosen from T1ce MR images and the other four slices were selected from the FLAIR MR images. The first column illustrates the bias-corrected MR images and the second column displays the provided ground truth maps. As shown in the first column,
Fig. 5. Comparison of the segmented tumor regions by the trained Deep-Net and the ground truth maps for eight exemplary slices from the test data are shown in this figure. The first four slices are $T_{1}ce$ MR images and the last four slices are FLAIR MR images. The last column displays the measured dice similarity index for the eight sample slices.
the extra background of the brain was cropped in each image to minimize the frequency difference between the inside brain and the background. The segmented regions by the trained Deep-Net are shown in the third column. The Deep-Net predicted maps were overlapped on the ground truth maps and the difference between the two maps is illustrated in column 4. The colors in the images of this column indicate the success or non-success performance of the trained network. The black, light, and dark gray shows the correct segmented pixels by the Deep-Net, the green pixels illustrate the pixels that the Deep-Net was not able to detect, and the purple color pixels are the pixels that are wrongly segmented by the Deep-Net. The last column of Fig. 5, the dice score, quantitate the similarity between the segmented regions by the Deep-Net and the GT maps. The exemplary slices in this figure visually imply that the trained Deep-Net successfully classifies the brain image pixels into the four main classes for both T1ce and FLAIR images. However, the Deep-Net can perform well for most tested slices, it has

Fig. 6. Six example slices from six different high-grade glioma patients to show Deep-Net defects in the classes segmentation.
other possible networks, such as Resnet50 and Xception which can be optimized model. The calculated Dice scores for ET, TC, and WT using achieved results, the median frequency balancing was replaced by in the segmentation of the classes. As can be observed in this figure, comprehensive comparison of three pre-trained networks: Resnet18, Resnet50 and Xception, to segment the different tumor regions.

Table 3

<table>
<thead>
<tr>
<th>Networks</th>
<th>Deeplabv3+</th>
<th>Deeplabv3+</th>
<th>Deeplabv3+</th>
<th>Deeplabv3+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Resnet50]</td>
<td>[Resnet18]</td>
<td>[Xception]</td>
<td>(Resnet18)</td>
</tr>
<tr>
<td>DSC for ET region</td>
<td>68.4%</td>
<td>78%</td>
<td>50.3%</td>
<td>77.4%</td>
</tr>
<tr>
<td>DSC for TC region</td>
<td>76.6%</td>
<td>85%</td>
<td>60.4%</td>
<td>84.5%</td>
</tr>
<tr>
<td>DSC for WT region</td>
<td>79.5%</td>
<td>87%</td>
<td>66.8%</td>
<td>85.5%</td>
</tr>
<tr>
<td>Validation Accuracy</td>
<td>96%</td>
<td>97.5%</td>
<td>93.2%</td>
<td>97%</td>
</tr>
<tr>
<td>Validation Loss</td>
<td>0.18</td>
<td>0.14</td>
<td>0.33</td>
<td>0.15</td>
</tr>
<tr>
<td>Run Time [min]</td>
<td>1741</td>
<td>274</td>
<td>683</td>
<td>196</td>
</tr>
</tbody>
</table>

| ET: Enhanced Tumor, TC: Tumor Core, WT: Whole Tumor. |
In this study, the average time needed to segment a brain slice by the constructed Deep-Net (on GPU, NVIDIA GeForce RTX 3080 with 12 GB RAM, in addition to CPU Intel Core i7-10700 K (@3.5 GHz) with 32 GB RAM on the Windows 10 system) is 0.03 sec, thus generation of the segmentation maps takes less than 5 sec for whole-brain examination with 155 slices. It seems that this time is acceptable and fast enough for an automatic segmentation algorithm, especially in contrast to the previously reported time for one slice segmentation. However, the reported time for different algorithms has been calculated on various hardware and computing platforms, so a decisive judgment between different models' speeds is a bit difficult. For example, Cui et al. [27] reported a time of 0.98 sec for one slice segmentation using an FCN-8s structure and 1.54 sec recruiting a two-stage algorithm. Both algorithms were executed on dual Intel E5-2603 CPUs and a middle-end GPU graphic card (GeForce GTX 1080, NVIDIA). Dong et al. [60] developed a deep algorithm that required 2-3 sec time for a slice segmentation. Their model was a U-Net-based fully convolutional network and performed on an NVIDIA Titan X Pascal GPU with 12G memory. Other researchers published computational times for their models such as 30 sec for a Multi-Scale 3D CNN with fully connected CRF on an NVIDIA GTX Titan X GPU, 3G RAM [23], 25 sec for TwoPathCNN and 3 min for Input-CascadeCNN on an NVIDIA TITAN black GPU [28] and 8 min for a deep CNN on a GPU NVIDIA GeForce GTX 980 equipped on an Intel Core i7 3.5 GHz [20], to predict a complete tumor study. Thus, it can be concluded that our Deep-Net is also time-efficient, besides its acceptable performance. However, a few previously constructed networks [27,60,61] reported slightly better segmentation performance than Deep-Net, but they are slower than the proposed algorithm in this work. The DSC result of this algorithm for enhancing tumor, tumor core and the whole tumor was compared with other works in order to show how this algorithm may improve HGG brain tumor segmentation performance. Table 4 summarized the DSC score of some recently reported constructed deep networks. All listed works in the table are trained by the MR images of the HGG patients. The Deep-Net was trained using only two more informative MRI sequences while others in the table were trained with three or four different MRI sequences. For example, Ahmad et al. [22] used a structure based on the ASPP and 3D U-Net and their reported DSC results are approximately the same as the obtained results for Deep-Net, despite using a 2D CNN structure and only two MRI sequences data in our model instead of 3D CNN structure and four MRI sequences data. Similarly, Yi et al. [61] employed data of four MRI sequences and a 3D CNN for segmentation of HGG brain tumor, as listed in Table 4, their reported DSC for WT and ET classes is 2% more than the calculated score for Deep-Net, however, the Deep-Net outperforms it for TC class. In the literature, the association of T1 and FALIR images of the BraTS2020 dataset has been similarly employed [62] on a DeepLabv3+ with different baseline networks such as MobileNetV2. This research reports a global accuracy of 96%, which is lower than Deep-Net.

Table 4
Comparison of the results of this study with previous investigations

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary of method</th>
<th>Database</th>
<th>MRI Modality</th>
<th>Grade</th>
<th>Dice Score Whole Tumor</th>
<th>Tumor Core</th>
<th>Enhancing Tumor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref# [2]</td>
<td>Convolutional Neural Networks</td>
<td>BraTS 2013</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>84</td>
<td>74</td>
<td>69</td>
</tr>
<tr>
<td>Ref# [20]</td>
<td>A CNN with small 3 × 3 kernels</td>
<td>BraTS 2013</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>88</td>
<td>76</td>
<td>73</td>
</tr>
<tr>
<td>Ref# [34]</td>
<td>stacked denoising autoencoder (SDAE), A deep neural network that reconstructs its input</td>
<td>BraTS 2015</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>88</td>
<td>83</td>
<td>77</td>
</tr>
<tr>
<td>Ref# [3]</td>
<td>U-Net Based Fully Convolutional Networks</td>
<td>BraTS 2015</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>81</td>
<td>68</td>
<td>64</td>
</tr>
<tr>
<td>Ref# [60]</td>
<td>3-D Convolution</td>
<td>BraTS 2015</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>89</td>
<td>76</td>
<td>80</td>
</tr>
<tr>
<td>Ref# [61]</td>
<td>Deep Cascaded Neural Network</td>
<td>BraTS 2015</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>90</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>Ref# [27]</td>
<td>Deep neural network</td>
<td>BraTS 2016</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>78</td>
<td>76</td>
<td>79</td>
</tr>
<tr>
<td>Ref# [6]</td>
<td>Dense Fully Convolutional Neural Network</td>
<td>BraTS2017</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>84</td>
<td>83</td>
<td>80</td>
</tr>
<tr>
<td>Ref# [7]</td>
<td>Deep Fully Convolutional</td>
<td>BraTS2019</td>
<td>T1, T2, T1c, Flair</td>
<td>HGG</td>
<td>87</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>Ref# [22]</td>
<td>residual-dense ASP</td>
<td>BraTS2020</td>
<td>T1c, Flair</td>
<td>HGG</td>
<td>87</td>
<td>85</td>
<td>78</td>
</tr>
</tbody>
</table>
global accuracy (97.5%). Deep cascade CNN showed relatively better performance for WT class, whereas DeepNet performed better for ET and TC classes. Cui et al. [27] used a BraTS2015 dataset and a two-step structure. In the first step, the tumor region was localized in each MRI slice, and then in the next step a CNN with deeper architecture and smaller kernels was recruited to classify HGG brain tumor into multiple sub-regions. They obtained a DSC of 71%, 76% and 88% for ET, TC and WT, respectively. Marciniewicz et al. [33] recruited a cascade multimodal U-net on BraTS2018 dataset (combined HGG and LGG) and achieved a DSC of 75%, 81% and 89% for the above classes. Kermi et al. [31] used the BraTS2018 dataset (combined HGG and LGG) employing a CNN architecture similar to U-net with essential modification on it. They reached a DSC of 72%, 79% and 87% for these tumor sub-regions. As can be understood from the Table 4 and the previously reported scores, the DSC of the DeepNet in segmentation of ET, TC, and WT classes (78%, 85%, and 87%) show a promising result to ensure that the Deep-Net structure can be accurate and reliable enough to be used in glioblastoma segmentation studies [18].

Despite the acceptable performance of the Deep-Net, some defects are observed in the segmented maps in some slices of test data. These defects may occur for classes with small areas, which leads to a large amount of type I error for such regions, and the Deep-Net is not successful in the classification of the pixels in the correct class. These defects are more notable when the enhanced tumor area is very small and thin. Another possible error that may occur in the segmented maps is the generalization and integration of the pixels in the periphery of that class. This defect is mostly observed in peri-tumoral edema and tumor necrotic core.

Conclusion

The results and experiments presented in this paper describe an end-to-end system for multi-modal segmentation of glioblastoma tumor sub-regions, normal tissue, and background based on a deep learning algorithm. The main contributions made in this work are finding the optimal effects are more notable when the enhanced tumor area is very small and thin. Another possible error that may occur in the segmented maps is the generalization and integration of the pixels in the periphery of that class. This defect is mostly observed in peri-tumoral edema and tumor necrotic core.


