

A new Online Class-Weighting approach with Deep Neural Networks for image segmentation of Highly Unbalanced Glioblastoma Tumors

Mostefa Ben Naceur, Rostom Kachouri, Mohamed Akil, Rachida Saouli

▶ **To cite this version:**

Mostefa Ben Naceur, Rostom Kachouri, Mohamed Akil, Rachida Saouli. A new Online Class-Weighting approach with Deep Neural Networks for image segmentation of Highly Unbalanced Glioblastoma Tumors. International Work-Conference on Artificial Neural Networks (IWANN-Advances in Computational Intelligence), Jun 2019, Gran Canaria, Spain. 10.1007/978-3-030-20518-8_46 . hal-02172208

HAL Id: hal-02172208
https://hal.science/hal-02172208

Submitted on 3 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A new Online Class-Weighting approach with Deep Neural Networks for image segmentation of Highly Unbalanced Glioblastoma Tumors

Mostefa Ben naceur[1,2] *, Rostom Kachouri[1], Mohamed Akil[1], and Rachida Saouli[2]

[1] Gaspard Monge Computer Science Laboratory, A3SI, ESIEE Paris, CNRS, University Paris-Est, France
mostefa.bennaceur@esiee.fr, rostom.kachouri@esiee.fr, mohamed.akil@esiee.fr
[2] Smart Computer Sciences Laboratory, Computer Sciences Department, Exact.Sc, and SNL, University of Biskra , Algeria
rachida.saouli@esiee.fr

Abstract. The most common problem among image segmentation methods is unbalanced data, where we find a class or a label of interest has the minority of data compared to other classes. This kind of problems makes Artificial Neural Networks, including Convolutional Neural Networks (CNNs), bias toward the more frequent label. Thus, training a CNNs model with such kind of data, will make predictions with low sensitivity, where the most important part in medical applications is to make the model more sensitive toward the lesion-class, i.e. tumoral regions. In this work, we propose a new Online Class-Weighting loss layer based on the Weighted Cross-Entropy function to address the problem of class imbalance. Then, to evaluate the impact of the proposed loss function, a special case study is done, where we applied our method for the segmentation of Glioblastoma brain tumors with both high- and low-grade. In this context, an efficient CNNs model called OcmNet is used. Our results are reported on BRATS-2018 dataset where we achieved the average Dice scores 0.87, 0.75, 0.73 for whole tumor, tumor core, and enhancing tumor respectively compared to the Dice score of radiologist that is in the range 74% - 85%. Finally, the proposed Online Class-Weighting loss function with a CNNs model provides an accurate and reliable segmentation result for the whole brain in 22 seconds as inference time, and that make it suitable for adopting in research and as a part of different clinical settings.

Keywords: Online Class-Weighting · Weighted Cross-Entropy · Convolutional Neural Networks · Deep Learning · Glioblastoma Tumors · Image Segmentation.

1 Introduction and Related Work

Current state-of-the-art image segmentation in the field of Deep Learning are based on Convolutional Neural Networks, where in general we find a feature

* Corresponding author.

extractor with a bank of convolution layers (i.e. trainable parameters), then pooling layers to make the images less sensitive and invariant to small translations (i.e. resisting to local translation), the last stage in CNNs models is a classifier which classifies each pixel (or voxel) into one of many classes. Since 2012, the tool of CNNs model led to a big breakthrough in all computer vision applications such as image classification, segmentation, e.g. LeCun et al. [1], Krizhevsky et al. [2] also in object detection Redmon et al. [3]. In the field of medical image classification and segmentation, we can classify methods-based CNNs into two categories: 2D-CNNs and 3D-CNNs models. Many works are proposed in the first category such as Axel et al. [4], Pereira et al. [5], Chang [6], Ben naceur et al. [7], Zhao et al. [8] and Havaei et al. [9] who extended the previous work of Axel et al. [4], where these methods use 2 dimensional patches as an input to train and classify medical image data. On the other hand, in the second category, we can cite works such as Urban et al. [10], Kamnitsas et al. [11, 12], where these methods use 3 dimensional image patches as an input to CNNs models. In the context of our ongoing work [7], we have faced and solved many Deep learning issues: (1) we proposed new three CNNs architectures for brain tumor segmentation problem, (2) we proposed a non-parametric fusion function that merges different brain tumor predictions, acting as a voting function which elects only the tumor region with a high probability, (3) we proposed a new optimizer to overcome the problem of vanishing gradient; where the gradient's signal becomes almost zero (vanishingly small) at the front layers of deep CNNs model. The proposed optimizer adds a new bloc after each training phase and so on until we obtain the satisfied segmentation results in terms of Dice coefficient.
The issue of data imbalance is common across multi-label image segmentation tasks, where there are many proposed methods in state-of-the-art. In overall there are two categories of solutions to this issue: some methods try to mitigate this problem by proposing equal sampling of training images patches [13], [9], on the other hand, some methods propose a new loss functions: cross entropy-based median frequency balancing [14], cross entropy-based weight map [15], combination of sensitivity and specificity [16], asymmetric similarity loss function [17], and many others [18], [19], [20], [21], [22]. To our knowledge, the existing methods are usually a specific function dedicated for well-defined applications however, these methods could be efficient in its corresponding applications but do not work well in others.
The aim of this paper is to overcome the issue of classical loss functions and define a generic one. To this end, we propose a new method called Online Class-Weighting. Then, to evaluate the impact and the effectiveness of our new proposed Weighted Cross-Entropy loss function, we train a CNNs model with our Online Class-Weighting for the problem of fully automatic brain tumor segmentation. For achieving this goal, our main contributions are divided into two folds:

1. To address the unbalanced data issue, we present a new Online Class-Weighting method that is based on the Weighted Cross-Entropy loss function,

2. To demonstrate the Online Class-Weighting performance, we evaluate our proposition within a case study applied to a fully automatic brain tumor segmentation of Highly Unbalanced Glioblastoma tumors.

2 Proposed Online Class-Weighting Method

Our proposed Online Class-Weighting method is based on the Weighted Cross-Entropy loss function (see equation 3) which is used in most image-related applications [14] [15].

$$p_j = \frac{\exp^{z_j}}{\sum_{k=1}^{n} \exp^{z_k}} \qquad for\ j \in \{1, .., n\} \tag{1}$$

$$Loss(p, q)_j = -\frac{1}{n}(\times \sum_j q_j \times \log(p_j)) \tag{2}$$

$$Loss(p, q)_j = -\frac{1}{n}(\sum_j W_j \times q_j \times \log(p_j)) \qquad w_j \in [0, 1]\ \ \& \ \sum_{j=1}^{n} W_j = 1 \tag{3}$$

Where Loss $(q, p)_j$ is the loss function (i.e. equation 3) that represents the error between the estimated probability $p_j \in$ [0,1] (i.e. equation 1) and the ground truth class $q_j \in \{0,1\}$, n is the number of classes, $w_j \in$ [0, 1] is the weighting factor assigned to the class j.

In this work, to make the prediction of the estimated probability accurate and faster, we have used one-hot encoding, where this encoding gives all probabilities of the ground truth to one class (i.e. the correct class), and the other classes become zero, e.g. q = [1,0,0,0], this vector q indicates that the first class is the correct class. In this case, to calculate the overall error we need only one operation instead of many operations for all classes.

The estimated probability is computed using Softmax function (see equation 1) which is used in the last layer of Neural Network after a forward propagation. Thus, this function squash the output to become between 0 and 1. Then, these probabilities are fed into Weighted Cross-Entropy function (see equation 3), where this function defines a weighting factor $w_j \in$ [0, 1] assigned to the class j compared to classical Cross-Entropy loss function (see equation 2). The main issue of the Weighted Cross-Entropy loss function is how to find the weighting factors w_j. To solve this issue, we propose here a new method called Online class-weighting. It allows to find w_j for each class during the training phase by searching for the best parameters using the training rate progress of each class with respect to different training iterations. Indeed, Online Class-Weighting initializes the weighting factors based on some computed measurements such as an evaluation metric, or an error's change. At the beginning, it measures the performance by evaluating the training results with and without the initialization of each class. This evaluation helps to measure the training accuracy of each class. After that, a new training is launched while rewarding each class with an adapted weighting factor w_j until the algorithm reaches a defined epoch iteration. w_j is computed regarding to the previous accuracy of the corresponding

class. We note that our proposed Online Class-Weighting algorithm is generic and able to be applied with any number of classes and suitable for different possible weighting factor initialization, and evaluation metrics. In the following, the Online Class-Weighting algorithm describes the proposed method.

Algorithm 1: Online Class-Weighting

Data: training dataset of m instances $(X_{1:m}, Y_{1:n})$, where an input image $X_t \in \mathbb{R}^{d \times d}$, and a target image $Y_j \in \mathbb{N}^{d \times d}$, n is the number of labels and $j \in \{1, ..., n\}$. Z, S are respectively the total number of epoches and a set of epoches. Z, S, $\in \mathbb{N}^+$, where $0 < S < Z$.

Result: set of weighting factors $\{w_j, / \sum_{j=1}^{n} w_j = 1, w_j \in [0, 1]\}$, and the final evaluation results M_j for each class j.

1 Initialization of the n weighting factors $w_{1,j}$, where $w_{1,j} = 1$;
2 Launch training for S epoches with the initialized weighting factors $w_{1,j}$;
3 Evaluation of the first training results for each class $(M_{1,j})$;
4 Weighting factors re-initialization $w_{2,j}$ where $w_{2,j} = \mathrm{f(X)}$, $\sum_{j=1}^{n} w_{2,j} = 1$;
5 Launch second training for S epoches with the initialized $w_{2,j}$;
6 Re-evaluation of the second training results for each class $(M_{2,j})$;
7 Computing stopping criteria: $C = \mathrm{g}(\mathrm{Error}_2(\text{training}), \mathrm{Error}_2(\text{testing}))$;
8 Initialization of the training iteration index: $i = 3$;
9 **while** *stopping criteria is not satisfied* **do**
10 Updating the weighting factors: $w_{i,j} = h(M_{i-1,j}, M_{i-2,j})$, where $\sum_{j=1}^{n} w_{i,j} = 1$ and i is the iteration index of the current S epoches;
11 Launch training with the updated weighting factors $w_{i,j}$;
12 Re-evaluation of the training results for each class $(M_{i,j})$;
13 Computing stopping criteria: $C = \mathrm{g}(\mathrm{Error}_i(\text{training}), \mathrm{Error}_i(\text{testing}))$;
14 Training iteration index i equals to: $i = i + 1$;

The algorithm of Online Class-Weighting works as a policy for the assessment of a Deep Learning model performance. Lines from 1 to 6 attempt to evaluate the behavior of a Deep learning model without and with applying the weighting factors. This assessment consists of computing the evaluation metrics (e.g. Dice coefficient, area under precision-recall curve [17]) of the training results for each class, for example Healthy, Necrosis and Non-enhancing tumor, Edema, and Enhancing tumor class. After that, we use these measurements to calculate the change in these weighting factors over the training phase. Line 7 computes the stopping criteria based on the training and testing errors. Lines 10 and 11 calculate the new weighting factors based on the evaluation metrics between the current training iteration and the two previous ones. To demonstrate the Online Class-Weighting performance, a case study for Glioblastoma tumor segmentation is described in the next section.

3 A case study: Online Class-Weighting approach for Glioblastoma tumor segmentation

In this section, we evaluate the proposed Online Class-Weighting algorithm with our CNNs model on BRATS-2018 dataset. For that, we present the used CNNs architecture that is designed for the segmentation of Glioblastoma brain tumors. Moreover, we introduce simple but effective settings (e.g. the weighting factors initialization) used in the Online Class-Weighting algorithm. Finally, we illustrate different experiments applied with different evaluation metrics.

3.1 Used Neural Networks Architecture

To demonstrate the performance and the efficiency of our loss function, we trained a fully convolutional neural networks architecture (see figure 1), where the design of this network is based on:

1. The extraction of multiple hierarchical representations (i.e. multiple feature maps) over the entire input image.
2. It is known that the use of pooling layer allows images to lose some information, which leads to reducing the features' size, thus we designed this network architecture to use the minimum number of pooling layers.
3. Instead of using fully connected layers as the default classifier in many CNNs architectures [1], [2], [23], we used a 1 x 1 x 4 convolution Softmax layer to save the memory and to speed up the inference time.
4. Our used CNNs architecture called OcmNet which is built based on the rule of using many interconnected modules, where this technique is implemented in known CNNs architectures such as GoogLeNet [24].

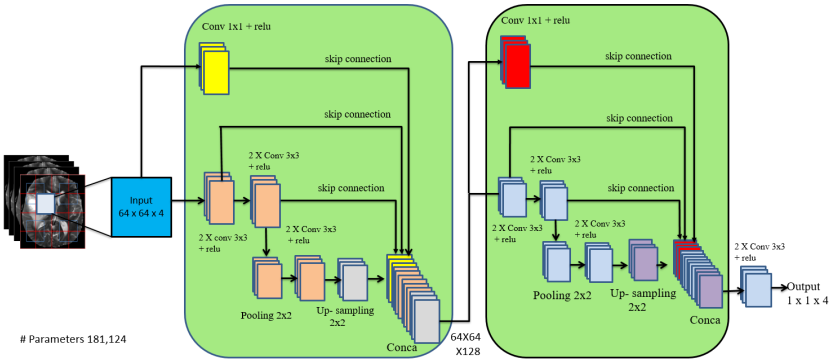


Fig. 1. The OcmNet architecture with 64 x 64 x 4 four channel input patches, consisting of two dense modules. The output 1 x 1 x 4 is a 1 x 1 convolution layer with 4 classes of BRATS dataset where we have 4 sub-regions: Healthy tissue, Necrosis and Non-enhancing tumor, Edema, Enhancing core. Also, this architecture has 181,124 parameters.

Convolutional Neural Networks are known for its ability to extract many complicated and hierarchical features from input images, but one of the issues known in CNNs algorithm is: it labels each pixel (or voxel) separately from the others (Havaei et al. [9], Pereira et al. [25]). Thus, to make CNNs take into account the influence of pixel's neighborhood, we concatenate the output of

different levels from the first module (see figure 1) with the second module to encourage the OcmNet to make a second prediction of the pixel's label. For the pre-processing step, we apply three normalization steps as follows:

1. Removing the 1% highest and lowest intensities: this technique helps to remove some noise at the tail of the histogram.
2. Subtracting the mean and dividing by the standard deviation of non zero values in all channels.
3. Isolating the background from the tumoral regions by assigning the minimum values to -9. This normalization step helps CNNs to differentiate easily between the background and the tumoral regions.

For the post-processing step, we remove some mis-classified non-tumor regions by applying two post-processing techniques:

1. Using a global threshold equals to 110 for each 2D MRI image to remove small non-tumoral regions based on connected-components. We refer to this Post-processing as Post-processing 1.
2. Using a Morphological opening operation, where we noticed that this operator improves the results of the first post-processing. We refer to this Post-processing as Post-processing 2.

3.2 Training and testing datasets

BRATS's dataset has a training set of 210 patients with high-grade Glioblastoma and 75 low-grade Glioblastoma. Glioblastoma tumors have 4 classes: Healthy tissue in addition to 3 sub-regions (Necrosis and Non-Enhancing tumor, Edema and Enhancing tumor). Each patient's brain image comes with 4 MRI sequences (i.e., T1, T1c, T2, flair) and the Ground truth labels that are made by radiologist. We split BRATS dataset into 70% for training (i.e. first phase), 30% for testing (i.e. second phase), then in the validation phase (i.e. third phase), we used BRATS 2018 validation set which contains 66 MRI images of patients with unknown grade. For the evaluation of our CNNs architectures on the validation dataset, we used the online evaluation system[3]. Moreover, OcmNet model is trained from scratch using a large number of MRI Overlapping Patches where we applied 25% overlap among patches.

3.3 Online Class-Weighting settings

After the definition of the Online Class-Weighting algorithm (see section 2), here we introduce different settings used in this algorithm:

1. Starting from the weighting factors: we initialize the weighting factors using the mean distribution, i.e. each weighting factor equals to $1/n$ (n is the number of classes), where n in our case study equals to 4, i.e. Healthy regions, Necrosis and Non-enhancing tumor, Edema and Enhancing tumor. The algorithm of Online Class-Weighting, line 4 (see algorithm 1), initializes the weighting factors $w_{2,j}$ as $w_{2,j} = f(X)$, thus $w_{2,j} = 1/n = 1/4 = 0.25$.

[3] Center for Biomedical Image Computing and Analytics University of Pennsylvania, Url:https://ipp.cbica.upenn.edu/

2. The second point consists of selecting the best candidates for the current iteration (i) during the training phase[4] as a function of distance between the training results of the iteration $(i-1)$ and the iteration $(i-2)$. After each S epoches (we found $S = 5$ provides the best results in our experiments) the algorithm changes the weighting factor according to $w_{i,j} = h(M_{i-1,j}, M_{i-2,j})$ (see algorithm 1, line 10), where $M_{i-1,j}$ and $M_{i-2,j}$ are the Dice coefficient (see section 3.4) of the training iteration $(i-1)$ and $(i-2)$ respectively. The function h selects two candidates in the following formulas then in the next step (see step 3), calculates the new weighting factors:

$$Candidate(class)_u = \underset{j\in[1:n]}{Argmax}(M_{i-1,1} - M_{i-2,1}), ...(M_{i-1,4} - M_{i-2,4)} \quad (4)$$

$$Candidate(class)_v = \underset{j\in[1:n]}{Argmin}(M_{i-1,1} - M_{i-2,1}), ...(M_{i-1,4} - M_{i-2,4)} \quad (5)$$

The algorithm chooses two candidates using $Argmax$ and $Argmin$ functions, where equation (4) returns the class that showed improvement for the last S epoches, thus the algorithm rewards this class. Equation (5) returns the class that is stucked in a valley of the sub-optimal point at the local mimina of loss function. In general, this class is the more frequent label, thus the algorithm penalizes this class.

3. After selecting the best candidates using $Argmax$ and $Argmin$, the algorithm computes the new weighting factors as the following:

$$W_{i,u} = W_{i-1,u} + \beta \qquad W_{i,v} = W_{i-1,v} - \beta \quad (6)$$

Where $W_{i,u}$, and $W_{i,v}$ in equation (6) are the best candidates, β is the rate of weight's change, where β in our experiments is equal to 0.2 for obtaining a soft weight's change between two successive weighting factors.

4. The last point is stopping criteria (see algorithm 1, lines 7 and 13):
   1. If the error's change (EC) between training and testing sets is greater than a threshold equals to 0.1: $EC =$ | Error(testing) - Error(training) |.
   2. Or testing set accuracy does not improve for a number of epoches K. We found $K = 20$ provides the best results in our experiments, (see figure 2).
   3. The last parameters is: if the above two criteria (i.e. a and b) are always false, thus the algorithm will stop after a defined number of epoches Z, where in our case study $Z = 240$.

3.4 Evaluation metrics

For the evaluation of our tumor segmentation method, we use the most pertinent evaluation metrics in state-of-the-art that are used in BRATS[5]: complete (i.e., necrosis and non-enhancing tumor, edema, enhancing tumor), core (i.e. necrosis and non-enhancing tumor, enhancing tumor), enhancing (i.e. enhancing tumor), the evaluation metrics are calculated as follows:
Dice= $\frac{|P_1 \wedge T_1|}{(|P_1| + |T_1|)/2}$, Sensitivity = $\frac{|P_1 \wedge T_1|}{|T_1|}$, Specificity= $\frac{|P_0 \wedge T_0|}{|T_0|}$,

[4] From this technique we inspired the name of Online Class-Weighting loss function
[5] https://www.med.upenn.edu/sbia/brats2018/data.html

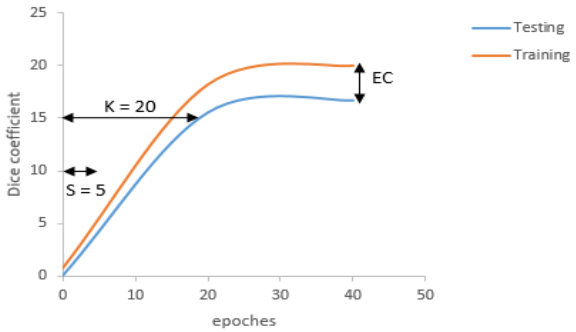


Fig. 2. The curve of stopping criteria that is based on three parameters : Error's change (EC), K epoches, S epoches.

$$\text{Hausdorff} = max\left\{ \sup_{p\in\partial P_1} \inf_{t\in\partial T_1} d(p\ ,\ t)\ ,\ \sup_{t\in\partial T_1} \inf_{p\in\partial P_1} d(t\ ,\ p) \right\}$$

Where $\wedge$ is the logical AND operator, P is the model predictions and T is the ground truth labels. T_1 and T_0 represent the true lesion region and the remaining normal region respectively. P_1 and P_0 represent the predicted lesion region and the predicted to be normal respectively. $(|.|)$ is the number of pixels. In addition, $'p'$ and $'t'$ are two points of the surface ∂P_1 and ∂T_1 respectively, and $d(p,t)$, $d(t,p)$ are the shortest least-squares distance (i.e. euclidean distance) between point $'p'$ and $'t'$ and vice versa for $d(t,p)$.

3.5 Experimental results and Discussion

Online Class-Weighting versus Cross-Entropy, Focal loss function: To demonstrate the performance of the Online Class-Weighting algorithm, we compare our loss function with state-of-the art Cross-entropy and Focal loss [21]. Cross-entropy and Focal loss are known and showed a high performance in many computer vision applications. Table 1 shows the results of our proposed CNNs architecture with 3 loss functions: Focal loss, Cross-Entropy and Online Class-Weighting. As we can see from table 1, according to the Dice score of Healthy, NCR/NET and Edema classes, we have competitive results with Cross-Entropy function. However, according to the Dice Complete, Core and Enhancing score, our OcmNet provides the best results with Online Class-Weighting loss. Thus, OcmNet architecture with Online Class-Weighting provides the best results on 5 over 6 metrics. With these results, Online Class-Weighting method outperformed the results of Cross-Entropy and Focal loss function.

Comparison to State-of-the-art: Table 2 illustrates the results of the segmentation performance of different methods. As we can see, our proposed OcmNet with Online Class-Weighting model improved the segmentation results in terms of Dice, sensitivity, specificity, Hausdorff distance. Moreover, OcmNet with Online Class-Weighting performs better than the latest methods applied to Glioblastoma brain tumor segmentation such as [4], [8], [12] in terms of Dice score and Specificity. Also, OcmNet with Online Class-Weighting architecture obtained competitive results with [5], [6], [9], [10] in terms of Dice coefficient and Hausdorff distance. Also, as we can see from table 2 and according to the standard deviation, OcmNet with Online Class-Weighting could achieve 100%

Table 1. Healthy, Necrosis and Non-enhancing tumor (NCR/NET), Edema, Enhancing tumor represent the different Glioblastoma tumor sub-regions segmentation results of OcmNet architecture with 3 different loss functions: Focal loss, Cross-Entropy, and our proposed Online Class-Weighting. Moreover, for each experiment we show the Dice score of 10 MRI images from BRATS-2018 dataset. Values in bold are the best results.

Methods	Dice score					
	Healthy	NCR/NET	Edema	Enhancing	Complete	Core
OcmNet + Focal loss	0.99	0.38	0.70	0.60	0.79	0.62
OcmNet + Cross-Entropy loss	0.99	**0.63**	0.80	0.72	0.87	0.81
OcmNet + Online Class-weighting	**0,99**	0,61	**0,80**	**0,75**	**0,88**	**0,83**

prediction on some patients MRI images in terms of Dice score and specificity, i.e. the prediction of OcmNet with Online Class-Weighting model corresponds to the manual segmentation of radiologist experts. Havaei et al. [9], Zhao et al. [8]

Table 2. Segmentation results of our proposed OcmNet and Online Class-Weighting method with the state-of-the-art CNNs methods. WT, TC, ET denote Whole Tumor (complete), Tumor Core, Enhancing Tumor core respectively. (±) is the standard deviation. Post 1 and Post 2 denote post-processing 1 and post-processing 2 respectively. Fields with (-) are not mentioned in the published work.

Methods	Dice score			Sensitivity			Specificity			Hausdorff		
	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET
Urban et al. [10]	0.88	0.83	0.72	-	-	-	-	-	-	-	-	-
Axel et al. [4]	0.79	0.68	0.57	-	-	-	0.79	0.67	0.63	-	-	-
Pereira et al. [5]	0.87	0.73	0.68	-	-	-	0.86	0.77	0.70	-	-	-
Chang [6]	0.87	0.81	0.72	-	-	-	-	-	-	9.1	10.1	6.0
Havaei et al. [9]	0.88	0.79	0.73	0.89	0.79	0.68	0.87	0.79	0.80	-	-	-
Kamnitsas et al. [12]	0.847	0.67	0.629	-	-	-	0.876	0.607	0.662	-	-	-
Zhao et al. [8]	0.84	0.73	0.62	-	-	-	0.82	0.76	0.67	-	-	-
OcmNet + Online Class-weighting	0.863 (±0.1)	0.752 (±0.2)	0.71 (±0.3)	0.863 (±0.2)	0.763 (±0.3)	0.774 (±0.3)	0.992 (±0.01)	0.996 (±0.01)	0.998 (±0.003)	23.37 (±28.2)	18.55 (±23.7)	11.97 (±24.6)
OcmNet + Online Class-weighting + Post 1	0.87 (±0.1)	0.753 (±0.2)	0.73 (±0.3)	0.86 (±0.2)	0.76 (±0.3)	0.776 (±0.3)	0.994 (±0.01)	0.996 (±0.01)	0.998 (±0.004)	14.71 (±22.5)	16.33 (±21.8)	9.83 (±20.7)
OcmNet + Online Class-weighting + Post 1 + Post 2	0.864 (±0.1)	0.75 (±0.2)	0.72 (±0.3)	0.853 (±0.2)	0.76 (±0.3)	0.76 (±0.3)	0.994 (±0.01)	0.996 (±0.01)	0.998 (±0.003)	14.63 (±22.4)	16.37 (±21.9)	8.14 (±20.04)

proposed a method based on equal sampling of training images patches to solve the problem of class-imbalance. However, we tried many experiments on this method and we found that it mitigates the impact of class-imbalance problem but it does not solve it (class-imbalance). Thus, loss function-based methods are the best choice and more robust. Cross entropy-based median frequency balancing [14], cross entropy-based weight map [15] propose a modified cross-entropy loss function, but each of which adapts this function to a specific application, which is not the case with our proposed Online Class-Weighting loss function. It does not assume any distribution or *prior knowledge* about the application fields before the training phase. In addition, the most important metric for a clinical decision support system [17] is sensitivity (recall), where OcmNet with Online Class-Weighting achieved a high accuracy on this metric (i.e. sensitivity), see table 2. In summary, OcmNet with Online Class-Weighting model obtained high

segmentation results, and so did with the Post-processing techniques, where the Post-processing 2 improved a lot the Hausdorff distance metric. The first conclusion from these segmentation results is that Gioblastoma brain tumors contain in most cases one connected region, i.e. we have demonstrated this hypothesis after applying the Post-processing 1, where we have seen an improvement in the segmentation performance. The second conclusion is that the class of Enhancing tumor does not have much border with the healthy tissue, i.e. this is demonstrated after applying the Post-processing 2, where we have seen a decrease in the surface of mis-classified Enhancing tumor region and an improvement in the Hausdorff distance metric. Finally, in this study, we have demonstrated that our proposed method has achieved the state-of-the-art segmentation performance in terms of Dice score, sensitivity, specificity, Hausdorff distance.

Inference time: To deploy a deep learning model on a large scale or on a real-time system, it is necessary to improve the deep learning inference time. Figure 3 shows the speed (seconds) versus CNNs-based methods. Our proposed OcmNet with Online Class-Weighting method is the fastest method among all other models.

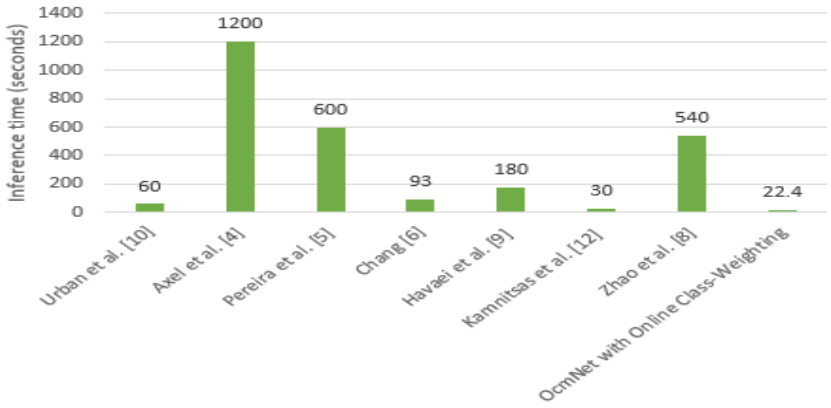


Fig. 3. Evaluation results of inference time (seconds) with the state-of-the-art CNNs methods versus our proposed OcmNet with Online Class-Weighting model.

4 Conclusion

In this work, we presented a new generic Online Class-Weighting algorithm, where the performance and the efficiency of this algorithm is demonstrated through a brain tumor segmentation case study with OcmNet model. In addition, the training of OcmNet with the Online Class-Weighting showed a high performance against Cross-Entropy and Focal loss functions. Moreover, the advantage of Online Class-Weighting is: it allows to find the weighting factors for each class during the training by searching for the best parameters using the rate of progress of each class with respect to the training phase. Moreover, our experimental results show that the OcmNet with Online Class-Weighting architecture improved the segmentation results. Also, these results are demonstrated through a separated validation dataset with real MRI images where the average Dice scores are 0.87, 0.75, 0.73, for whole tumor, tumor core, and enhancing

tumor respectively. OcmNet with Online Class-Weighting and Post-processing 1 and Post-processing 2 is the best architecture in terms of Dice, sensitivity, specificity and Hausdorff distance. This architecture has only 181,124 parameters and that make it suitable for adopting in research and as a part of different clinical settings. In this work, we proposed a generic loss function that can be applied to other applications. Based on the preliminary results, our OcmNet with Online Class-Weighting algorithm provides a very promising result. As a perspective of this research, we intend to investigate different scenarios and parameters for an optimal use of the proposed Online Class-Weighting method.

References

1. Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
2. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
3. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
4. Axel Davy, Mohammad Havaei, David Warde-Farley, and T. Lam J. Pierre-Marc C. Aaron L. Hugo P. Chris B. Yoshua Biard, Antoine. Brain tumor segmentation with deep neural networks. *in: Proceedings of the MICCAI Workshop on Multimodal Brain Tumor Segmentation Challenge BRATS*, pages 01–05, 2014.
5. Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Deep convolutional neural networks for the segmentation of gliomas in multi-sequence mri. *in: Proceedings of the MICCAI Workshop on Multimodal Brain Tumor Segmentation Challenge BRATS*, pages 52–55, 2015.
6. Peter D Chang et al. Fully convolutional neural networks with hyperlocal features for brain tumor segmentation. In *Proceedings MICCAI-BRATS Workshop*, pages 4–9, 2016.
7. Mostefa Ben naceur, Rachida Saouli, Mohamed Akil, and Rostom Kachouri. Fully automatic brain tumor segmentation using end-to-end incremental deep neural networks in mri images. *Computer methods and programs in biomedicine*, 166:39–49, 2018.
8. Xiaomei Zhao, Yihong Wu, Guidong Song, Zhenye Li, Yazhuo Zhang, and Yong Fan. A deep learning model integrating fcnns and crfs for brain tumor segmentation. *Medical image analysis*, 43:98–111, 2018.
9. Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.
10. Gregor Urban, M Bendszus, F Hamprecht, and J Kleesiek. Multi-modal brain tumor segmentation using deep convolutional neural networks. *MICCAI BraTS (Brain Tumor Segmentation) Challenge. Proceedings, winning contribution*, pages 31–35, 2014.
11. Konstantinos Kamnitsas, Enzo Ferrante, Sarah Parisot, Christian Ledig, Aditya V Nori, Antonio Criminisi, Daniel Rueckert, and Ben Glocker. Deepmedic for brain tumor segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 138–149. Springer, 2016.

12. Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.
13. Matthew Lai. Deep learning for medical image segmentation. *arXiv preprint arXiv:1505.02000*, 2015.
14. Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
15. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
16. Tom Brosch, Lisa YW Tang, Youngjin Yoo, David KB Li, Anthony Traboulsee, and Roger Tam. Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. *IEEE transactions on medical imaging*, 35(5):1229–1239, 2016.
17. Seyed Raein Hashemi, Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, Sanjay P Prabhu, Simon K Warfield, and Ali Gholipour. Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection. *IEEE Access*, 7:1721–1735, 2019.
18. Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.
19. Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.
20. Lucas Fidon, Wenqi Li, Luis C Garcia-Peraza-Herrera, Jinendra Ekanayake, Neil Kitchen, Sébastien Ourselin, and Tom Vercauteren. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. In *International MICCAI Brainlesion workshop*, pages 64–76. Springer, 2017.
21. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
22. Xuan Chen, Jun Hao Liew, Wei Xiong, Chee-Kong Chui, and Sim-Heng Ong. Focus, segment and erase: An efficient network for multi-label brain tumor segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 654–669, 2018.
23. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
24. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
25. Sérgio Pereira, Américo Oliveira, Victor Alves, and Carlos A Silva. On hierarchical brain tumor segmentation in mri using fully convolutional neural networks: a preliminary study. In *2017 IEEE 5th Portuguese meeting on bioengineering (ENBENG)*, pages 1–4. IEEE, 2017.