Fully automatic brain tumor segmentation with deep learning-based selective attention using overlapping patches and multi-class weighted cross-entropy

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

Fully automatic brain tumor segmentation with deep learning-based selective attention using overlapping patches and multi-class weighted cross-entropy

Mostefa Ben naceur[a,b,*], Mohamed Akil[a], Rachida Saouli[b], Rostom Kachouri[a]

[a] *Gaspard Monge Computer Science Laboratory, Univ Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France*
[b] *Smart Computer Sciences Laboratory, Computer Sciences Department, Exact.Sc, and SNL, University of Biskra, Algeria*

Abstract

In this paper, we present a new Deep Convolutional Neural Networks (CNNs) dedicated to fully automatic segmentation of Glioblastoma brain tumors with high- and low-grade. The proposed CNNs model is inspired by the Occipito-Temporal pathway which has a special function called *selective attention* that uses different *receptive field* sizes in successive layers to figure out the crucial objects in a scene. Thus, using *selective attention* technique to develop the CNNs model, helps to maximize the extraction of relevant features from MRI images. We have also treated two more issues: class-imbalance, and the spatial relationship among image Patches. To address the first issue, we propose two steps: an equal sampling of images Patches and an experimental analysis of the effect of weighted cross-entropy loss function on the segmentation results. In addition, to overcome the second issue, we have studied the effect of Overlapping Patches against Adjacent Patches where the Overlapping Patches show a better segmentation result due to the introduction of the global context as well as the local features of the image Patches compared to the conventionnel Adjacent Patches method. Our experiment results are reported on BRATS-2018 dataset where our End-to-End Deep Learning model achieved state-of-the-art performance. The median Dice score of our fully automatic segmentation model is 0.90, 0.83, 0.83 for the whole tumor, tumor core, and enhancing tumor respectively compared to the Dice score of radiologist, that is in the range 74% — 85%. Moreover, our proposed CNNs model is not only computationally efficient at inference time, but it could segment the whole brain on average 16 seconds. Finally, the proposed Deep Learning model provides an accurate and reliable segmentation result, and that makes it suitable for adopting in research and as a part of different clinical settings.

Keywords: Brain Tumor Segmentation, Class-imbalance, Convolutional Neural Networks, Fully Automatic, Glioblastomas, Overlapping Patches.

*Corresponding author.
Email addresses: mostefa.bennaceur@esiee.fr (Mostefa Ben naceur), mohamed.akil@esiee.fr (Mohamed Akil), rachida.saouli@esiee.fr (Rachida Saouli), rostom.kachouri@esiee.fr (Rostom Kachouri)

1. Introduction

A brain tumor is a growing abnormal cell in the brain or central spin canal (Young and Knopp, 2006). According to the NBTS [1], in the United Kingdom, there are more than 4.200 patients with primary brain tumors (Logeswari and Karnan, 2009). In the United States of America, each year, 13.000 patients die, and 29.000 patients suffering from primary brain tumors (Singh et al., 2012). Moreover, each patient has a different health condition, age, gender, and a different tumor that could appear anywhere in the brain. With this huge number of patients in the world and the massive medical data such as device recording, magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, computed tomography (CT), X-rays, emerged the need to discover new techniques for the treatment of patients. In addition, the diagnosis period plays a very important role, especially when we have patients with life-threatening (Schneider et al., 2010) diseases such as Gliomas tumors, these tumors according to the World Health Organization classification (Gupta and Dwivedi, 2017; Louis et al., 2016), have 4 grades (I, II, III, IV). In addition, Gliomas affect children between 5 and 10 years (Goodman and Fuller, 2014), and adults between 40 and 65 years (Schneider et al., 2010). Moreover, Gliomas tumors are the most frequent primary brain tumors in adults (Holland, 2001), these tumors represent 81% of all malignant brain tumors (Ostrom et al., 2014), and 45% of all primary brain tumors (Liu et al., 2016). Some patients with Gliomas tumors have survival rate between 0.05% and 4.7% (Ostrom et al., 2014), representing with that the second reason of death. In this paper, our method is oriented to the segmentation of Glioblastomas (see figure 1) which are brain tumors belonging to the category of Gliomas tumors.

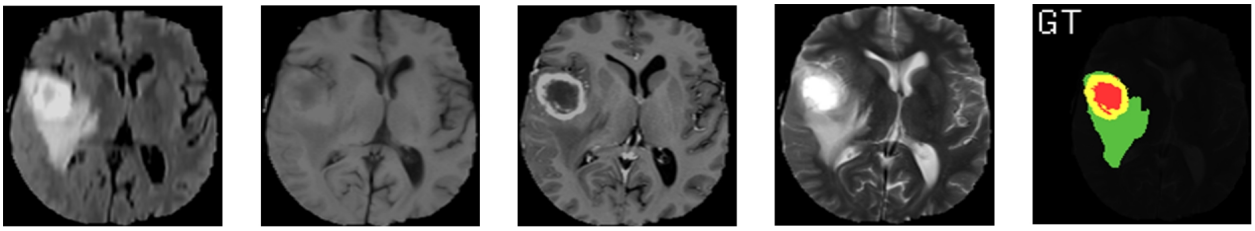


Figure 1: The four multi-sequence MR images of patients with Glioblastomas tumors. From left to right: Flair, T1, T1c, T2, Ground Truth (GT) labels (created by expert). The color is used to distinguish between the tumor regions: red: Necrotic and Non-Enhancing tumor, green: Peritumoral Edema, yellow: Enhancing tumor, black: Healthy tissue and background. Best viewed in color.

Usually, an expert radiologist uses MRI as the most effective (Akram and Usman, 2011; Bhandarkar and Nammalwar, 2001) technique to generate Multi-modal images to identify different tumor regions in the soft tissues. Moreover, the radiologist generates four standard MRI images modalities for Gliomas diagnosis (Işın et al., 2016): T2-weighted fluid attenuated inversion recovery (Flair), T1-weighted (T1), T1-weighted

[1] NBTS: National Brain Tumor Society

contrast-enhanced (T1c), and T2-weighted (T2) for each patient, then each region of Gliomas tumors needs to be segmented pixel-by-pixel in each slice until the 3D brain volume is divided into meaningful regions as shown in the Ground Truth image (see figure 1). After that, this segmentation map is used for treatment sessions, surgery planning, and follow-up to see if the tumor is growing or shrinking. This procedure from generating Multi-modal MR images, segmentation, diagnosis, treatment, and follow-up, takes at least 2 months and around 5% of patients stay under this procedure for 5 years (Ostrom et al., 2014).

To interpret brain MRI images, a specialized radiologist employs a manual segmentation using information from MRI images with anatomical and physiological knowledge (Işın et al., 2016). Where it's known that the manual segmentation in MRI images is a time-consuming and a tedious procedure (Abd-Ellah et al., 2016). The key challenge is the interpretation and the segmentation of brain MRI images that depend on the expertise of each radiologist (Akram and Usman, 2011). Moreover, the mission of radiologist becomes very difficult in the case if the tumor region has intra-tumoral structures such as the Glioblastomas tumors which have three different structures besides healthy tissue inside the tumor region: Necrotic and Non-Enhancing tumor, Peritumoral Edema, Enhancing tumor (see figure 1). That's why the Dice score of human rater (i.e. radiologists) is in the range 74% — 85% (Menze et al., 2015). In addition, the radiologist spends between 3 and 5 hours (Kaus et al., 2001) to extract features manually from MRI images then to label each pixel if it belongs to a tumor region and which region from 3 regions or it belongs to healthy tissue. Thus, the manual segmentation is a time-consuming compared to automated methods which take between few seconds and 20 minutes using GPU implementation. Since the aim is to obtain a segmentation of brain tumors, given the number of patients, obviously fully automated segmentation methods are practical for day-to-day use in clinical centers and for research.

After the breakthrough in 2012 in the field of computer vision where a team developed a deep learning model called AlexNet (Krizhevsky et al., 2012), this model outperformed the state-of-the-art methods and it achieved the best result in the field of object recognition. Since 2014, many deep learning-based research (Ben naceur et al., 2018; Havaei et al., 2017; Pereira et al., 2015) have been proposed in the field of brain tumor segmentation. Also, in the last decades, we noticed that the number of published papers involving brain tumor segmentation increased exponentially (Menze et al., 2015), this increase indicates that creating a Computer-Aided Diagnosis (CAD) (Akram and Usman, 2011; El-Dahshan et al., 2014) has always been a highly needed option and necessary (Zhang et al., 2001). In addition, some patients have aggressive tumors that need to be treated in less than two months. This CAD system could decrease the needed time for the diagnosis process (El-Dahshan et al., 2014) and it could give to Oncologists more time with their patients in the process of treatment and follow-up.

The addressed problem in this paper is how to obtain the most accurate and reliable brain tumor segmentation in the light of many challenges: the most important one is (i) Glioblastomas tumors have 3 sub-regions

in addition to Healthy tissue: Necrotic and Non-Enhancing tumor, Peritumoral Edema, Enhancing tumor, where also Gliomas, including Glioblastomas, invade the surrounding tissue rather than displacing it, causing unclear boundaries, moreover, Gliomas in MRI images have the same appearance as Gliosis, stroke, inflammation, blood spots (Goetz et al., 2016), to solve this problem, we propose a CNNs model that is inspired by the Occipito-Temporal Pathway (OTP), this structure made up of many sub-regions (V1, V2, V4, IT) (Desimone et al., 1989). The design of our CNNs model allows to maximize the features' representation of Glioblastomas tissue, also leads to detect and predict even small regions with fuzzy borders.

(ii) The second challenge is the Artificial Neural Networks, including CNNs, do not perform well with highly unbalanced data such as BRATS dataset where 98% of data (Havaei et al., 2017) are healthy tissue (i.e. non-tumorous). The training of CNNs architecture with unbalanced data, will make predictions with low sensitivity, thus the CNNs architecture will bias toward the healthy class. In medical applications, the most important metric for a clinical decision support system is sensitivity (Hashemi et al., 2018) toward tumoral regions. Thus, to overcome this problem, we conduct many class-weighting evaluations (see section 3.4) by adding weights to the loss function. Like this, we can adjust the contribution of each output by a specific weight, which has been shown to be effective for image segmentation (Badrinarayanan et al., 2015) and for the problem of unbalanced medical data such as brain tumor segmentation (see section 3).

(iii) The third challenge is common in semantic image segmentation problem, where this challenge is about the reduction of features during the pooling layers and convolution striding (Chen et al., 2017). To treat this challenge, we use short and long skip connections (Drozdzal et al., 2016; He et al., 2016) which encourage the reuse of low-level features such as lines, edges. The combination of low-level features with high-level features such as shapes, objects helps CNNs architectures to better locate the boundaries of tumor regions.

(iv) The fourth challenge is to make the prediction more accurate by solving the limitation of segmentation methods that are based on image Patches; it's known that CNNs do not take into account the relationship among image Patches (Zhao et al., 2018). In our work, Overlapping Patches is used to extract more Patches in the intersection of Adjacent Patches. Like this, we can extract global features such as the relationship among Patches and their positions in the entire brain image, as well as local features such as pixel intensity and pixel label. Through this work, we developed an End-to-End framework for brain tumor segmentation based on a new inspired CNNs architecture. To the best of our knowledge, this work is the first work succeed to address these four challenges together.

In this paper, our goal is to propose and develop an End-to-End Deep Convolutional Neural Networks model for fully automatic brain tumor segmentation. The proposed model is used to segment the brain tumors of Glioblastomas with both high- and low-grade. For achieving this goal, we propose four contributions:

1. First we propose a new Deep Learning model with 3 variants architectures which give more accurate and reliable results, these architectures achieved a high segmentation performance compared to the

state-of-the-art methods. Details of these contributions are dedicated to address the first and the third challenges.

2. To overcome the unbalanced data problem, we studied the effect of class-weighting technique on CNNs segmentation results, where we propose a set of weights for the 4 sub-regions. Moreover, details of this technique in section 3.4 are oriented to treat the second challenge.
3. We propose also Overlapping Patches technique to improve the segmentation results, i.e. the use of Adjacent Patches help to extract local features, but the use of Overlapping Patches helps to use small Patches and to extract both local and global features. The Overlapping Patches allowed us to handle the fourth challenge (in section 3.2).
4. Our CNNs model is a very fast model, also it could segment the whole brain on average 16 seconds, i.e. reducing the number of parameters and using small Patches through Overlapping Patches helped to reduce the training and the inference time.

After reviewing the state-of-the-art in the field of brain tumor segmentation, machine learning based methods, and the current CNNs-based methods in section 2. Then, we explain our approach for solving the problem of brain tumor segmentation in section 3. Then, we describe BRATS dataset and the evaluation metrics, also the results of our approach in section 4. Finally, the conclusion and the perspectives are described in section 5.

2. Related work

Most research on Pattern Recognition and Machine Learning from 1970s to 1990s: were based on features engineering (i.e. hand-designed features) and mathematical models, in which the researchers use training data to extract the input (binary or real) vector of features, then the research in 1990s to 2000s jumped to use less feature engineering and more discriminative models which use a lot of data to extract several features then using a classifier at the end of the pipeline to distinguish between the different classes. These classical approaches are in overall: threshold-based methods (Gibbs et al., 1996; Stadlbauer et al., 2004), Region-based methods (Cates et al., 2005; Kaus et al., 2001), Edge-based methods (Caselles et al., 1993; Lefohn et al., 2003), Atlas-based methods (Menze et al., 2010; Prastawa et al., 2003), Classification and Clustering methods (Bhandarkar and Nammalwar, 2001; Clark et al., 1998; Fletcher-Heath et al., 2001). These methods generally produce poor results and need a user interaction (e.g., region growing) most of the time and sensitive to noise (e.g., edge detector). Moreover, they need *prior knowledge* from experts and feature engineering. Also, they are in overall computationally expensive and use a lot of memory due to computing a huge number of features.

In 2006, appeared a new type of learning method called Deep Learning which uses a huge number of data to extract many low-features such as lines, edges with different orientations then it combines them in a

hierarchical way to obtain high-level features such as shapes, objects, and faces. Moreover, the research on tumor segmentation or lesion segmentation in overall with Deep Learning started in 2014.

(Havaei et al., 2017) proposed a fully automatic brain tumor segmentation method based on Cascaded Convolutional Neural Networks which are an extended version of the work (Axel et al., 2014), these Cascaded Networks used two pathways that are trained in different phases to capture local and global features. They used also as input 2D Axial Patches with four MR sequences as channels where each pathway has a different input patch size. Moreover, they proposed two stages of training for the problem of class-imbalance to correct in the second training stage the Patches that are biased toward the wrong class. After that, they applied a threshold technique as a post-processing method to remove the connected-components near to the skull. In addition, this Cascaded architecture takes 180 seconds to segment the complete brain using GPU implementation (Havaei et al., 2017).

(Chang, 2016) developed a CNNs model that is based on two concepts (1) a fully convolutional architecture that predicts a dense output matrix size as used in the original input (Long et al., 2015), (2) Hyperlocal features concatenation; the input MR images is re-introduced in the concatenation layer before the output. This technique is used first by (Yang and Ramanan, 2015) in their architecture Directed-Acyclic-Graph which is a new variant of standard CNNs (LeCun et al., 1998). The architecture of (Chang, 2016) has 7 convolution layers in addition to upsampling and concatenation layers. In this CNN architecture, 4 channels of MRI images are used as an input (i.e., Flair, T2, T1c, T1). Moreover, this architecture takes 0.93 seconds to segment the entire brain using GPU implementation (Chang, 2016).

(Ellwaa et al., 2016) proposed an iterative method which is based on a random forest with 100 trees, each of which has depth 45, their method extracts 328 features from MRI images, these features are gradient features, appearance features, and context-aware features. The input to their method is 4 channels of MRI images (i.e., FLair, T1, T1c, T2). This iterative method works by choosing in each iteration 5 patients and add them to the training set, then they continue the training of this random forest until the training set reaches 50 patients, where at 50 patients their iterative method stops.

(Kamnitsas et al., 2016, 2017) developed a 3D-CNNs model for brain tumor segmentation based on the model's performance (Urban et al., 2014). These 3D-CNNs networks composed of dual pathway with 11 layers, the input to this network is 3D MRI images (i.e. Flair, T1, T1c, T2), also each pathway has a different input patch size (i.e. $4 \times 25^3, 4 \times 19^3$). Then, they added conditional random field as a post-processing operation to remove mis-classification regions and as a spatial regularization. Moreover, they extended their network with residual connections (He et al., 2016), in which this new extended network (Kamnitsas et al., 2016) did not obtain a big improvement compared to the original model (Kamnitsas et al., 2017). Their 3D CNNs model takes 24 hours for training using GPU implementation, and for testing, it takes 35 seconds to segment the entire brain.

(Zhao et al., 2018) developed a brain tumor segmentation method based on the integration of CNNs and conditional random field in one network, as opposed to (Kamnitsas et al., 2017) who used conditional random field as a post-processing step. The authors of (Zhao et al., 2018) developed 3 CNNs networks that take as an input 3 types of MRI images (i.e., Flair, T1c, T2). Each of these 3 networks use two pathways similar to (Havaei et al., 2017; Kamnitsas et al., 2017), where these pathways are trained on 2D image Patches (i.e., 33×33 and 65×65) and slices (i.e. 240×240) from axial, coronal and sagittal views. For the testing step, the prediction results from 3 views are fused using a voting strategy. Moreover, these 3 networks took 12 days for training using GPU implementation, and for testing, each model took for each view (i.e., axial, sagittal, coronal) on average 3 minutes to segment the entire brain, i.e., 3 networks $\times$ 3 minutes in addition to fusion time which is not reported in the original paper.

In a previous work (Ben naceur et al., 2018), we proposed 3 fully automatic brain tumor segmentation methods, where we achieved the state-of-the-art performance. The proposed methods are based on the concept of incremental optimization; after one cycle of training, a new block is added automatically on top of last blocks. The followed strategy in this work is based on 2 rules (1) as we add more layers, we get a better segmentation performance, (2) after many experiments to form one block, we found that deep learning models that are based on 2 or 3 consecutive layers of convolution provide a good segmentation performance. This new strategy called automated machine learning (AutoML) (Zoph et al., 2018; Pham et al., 2018), is a new field of artificial intelligence; AutoML attempts to design new machine learning models without the intervention of users. The proposed segmentation methods are trained on 2D image Patches (32×32) that are extracted from the Axial view. The training time took ≈ 5 hours and the testing time took in range 19 - 21 seconds to segment the entire brain tumor using GPU implementation. The limitation of this method is the GPU memory; where after a certain training time, the memory gets overwhelmed by the added blocks. To overcome this issue, we replaced in this paper the added blocks by a new concept called modules. These modules are inspired by the interconnection among the brain's regions: retina, V1, V2, V4, IT.

(Mlynarski et al., 2019) developed a fully automatic brain tumor segmentation method based on a combination of six models of 3D Convolutional Neural Networks (CNNs) architectures. Each of these architectures is composed of three (or five) CNNs architectures, also, these architectures are trained independently and dedicated for one MRI view (e.g. Axial, Coronal or Sagittal slices of the input image) and one architecture based on 3D CNNs. The proposed segmentation 3D CNNs model is trained on channels concatenation between extracted feature maps from the Axial, Coronal and Sagittal dedicated architectures and two (T2 , T1c) or four (T2, T2-Flair, T1, T1c) 3D multisequence MR images. The technique of using feature maps as an additional input into the architecture of another CNNs, is used by (Havaei et al., 2017). Moreover, (Mlynarski et al., 2019) addressed two issues: the first one is long rang context, to solve this issue, they used 2D CNNs (its input is one of three views) to capture rich information through increasing the size of

the receptive field. The second issue is unbalanced data, where they solved this issue by using weighted cross-entropy as a loss function.

2.1. Related work discussion

As a summary of related work, the cited works (Chang, 2016; Ellwaa et al., 2016; Havaei et al., 2017; Kamnitsas et al., 2017; Zhao et al., 2018; Ben naceur et al., 2018; Mlynarski et al., 2019) address the main problem of fully automatic brain tumor segmentation in different ways. In addition to the main problem, there are three important subproblems: (1) unbalanced data; the class of interest (i.e. tumoral region) has the minority of labels which represent for example in our dataset almost 2% of data, while the healthy class has 98% of data. This subproblem is solved by two training step (Havaei et al., 2017), equal sampling of training patches (Havaei et al., 2017; Zhao et al., 2018), weighted cross-entropy (Mlynarski et al., 2019), Online Class-Weighting Approach (Ben naceur et al., 2019), (2) long-range context modeling (e.g., tumor shape, relationship between tumor parts, spatial dependencies) using all pixels in the same 2D slice, as opposed to the short-range 3D context that is computationally expensive. In this case, to achieve the trade-off and to benefit from 3D context through 3 views (i.e., Axial, Coronal, and Sagittal), thus using small 3D patches is the best solution. In images segmentation applications, short- and long-range context modeling could enhance CNN's capability to distinguish pixels, especially those at the borders. The subproblem of long-range context is solved by (Mlynarski et al., 2019), also this subproblem is solved partially by (Chang, 2016; Havaei et al., 2017; Zhao et al., 2018; Ben naceur et al., 2018). Moreover, one of the important sub-problems in CNNs is (3) the relationship among patches (also known as modeling the patches in images), because the training method typically used in CNN is patch-by-patch, we refer to this method as Adjacent Patches. The limit of Adjacent Patches is that it does not take into consideration (Havaei et al., 2017; Zhao et al., 2018) that the set of patches (e.g., each patch has 64 x 64) represents the full image (in our case 240 x 240). The issue of modeling both context and spatial relationship in image segmentation is solved by multi-scale (multi-resolution) approach (Kamnitsas et al., 2017), we also solved this issue by using our proposed Overlapping Patches approach (see section 3.2).

In this paper, we developed new CNNs architectures from scratch, these architectures are able to detect effectively the tumoral regions. Also, we addressed many issues such as unbalanced data and the spatial relationship among patches, also we solved partially the issue of long-range context. The first issue (i.e., unbalanced data) is addressed using the class-weighting technique, and the second issue is solved using Overlapping Patches technique. In addition, we choose the best four positions in a MRI image to extract the most relevant features with the minimum patches. The third issue (i.e. long-range context) is solved partially using Axial MRI images (i.e. 2D Axial slice) to reduce the computational cost of the CNN's parameters in addition to reduce the training and the inference time. Consequently, the solved issues allowed us to outperform the radiologist for the Dice score metric in addition to achieve the state-of-the-art performance.

In this study, we have been motivated by the success of deep learning in computer vision and in the context of brain tumor segmentation. From 2014 until now, the brain tumor segmentation performance has been increased a lot and the inference time of fully automatic methods has been decreased dramatically from ≈ 20 minutes to few seconds. Here besides the functionality of CNNs, we propose deep learning architectures inspired by the visual cortex structure to model the function and the structure of biological neural networks. These architectures are trained using MRI image Patches extracted from the axial view. Moreover, to avoid the class-imbalance problem, we apply two techniques: the number of training Patches is equally extracted for each class, and a new class-weighting technique is proposed. In addition, we have also studied the effect of Overlapping Patches on deep learning performance in order to solve the issue of brain tumor segmentation methods that are based on image Adjacent Patches. The novelty of our architectures is: (1) the efficiency in terms of inference time and in terms of memory by removing fully connected layers and replacing them by Fully Convolutional Networks (Long et al., 2015), (2) our analysis of the blocks technique (Ben naceur et al., 2018), led us to replace this technique by the concept of modules that showed an improvement of the boundary delineation. In addition, (3) the solved issues (i.e. unbalanced data, relationship among patches) allowed this pipeline to reach the state-of-the-art performance.

3. Proposed brain tumor segmentation method

The proposed brain tumor segmentation pipeline has three stages: pre-processing, then segmenting the Overlapping MRI Patches using CNNs-based architectures, and finally two post-processing techniques.

3.1. Pre-processing

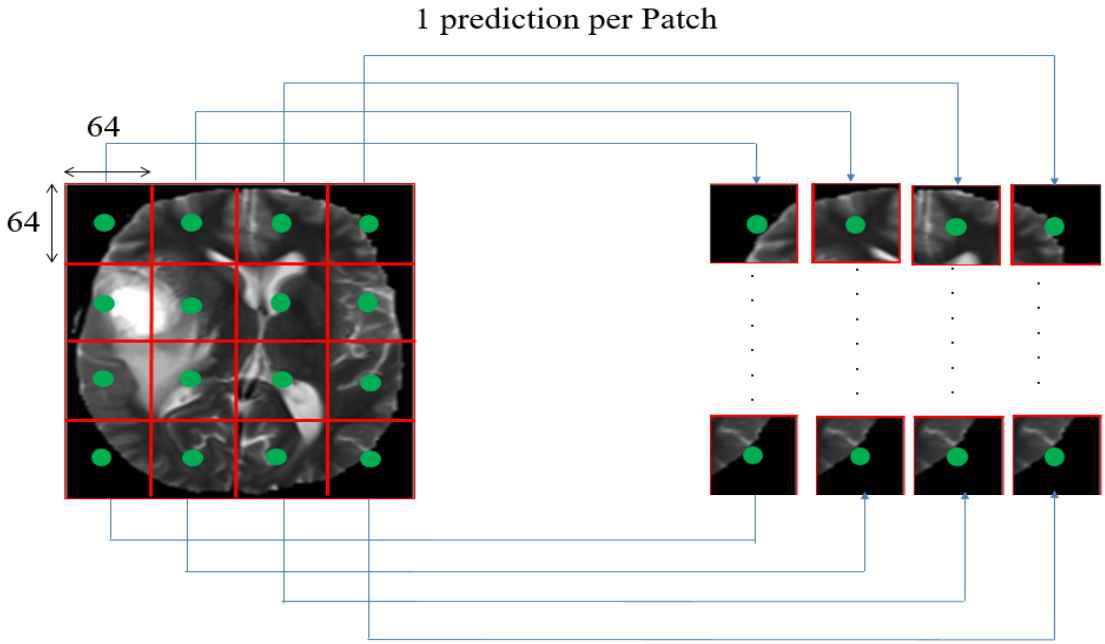
To remove some noises and to enhance the quality of MRI images, especially the MRI images of the BRATS dataset are generated using different MRI devices and acquisition protocols. Thus, to mitigate the issue of different intensity ranges among the BRATS's images, we applied 3 pre-processing steps based on this work, and on (Ben naceur et al., 2018). Our normalization method of each 2D axial image (i.e. a slice) is as follows:

1. Removing 1% highest and lowest intensities: this technique helps to remove some noises at the tail of the histogram, where this step has provided good results in many research (Havaei et al., 2017; Tustison, 2013).
2. Subtracting the mean and dividing by the standard deviation of non-zero values in all channels: this technique is used to centre and to put the data in the same scale, i.e., bringing the mean intensity value and the variance between one and minus one.
3. In this step, we try to isolate the background from the tumoral regions by assigning the minimum values to -9, where it has been observed that using integer numbers between -5 to -15, fits our CNNs

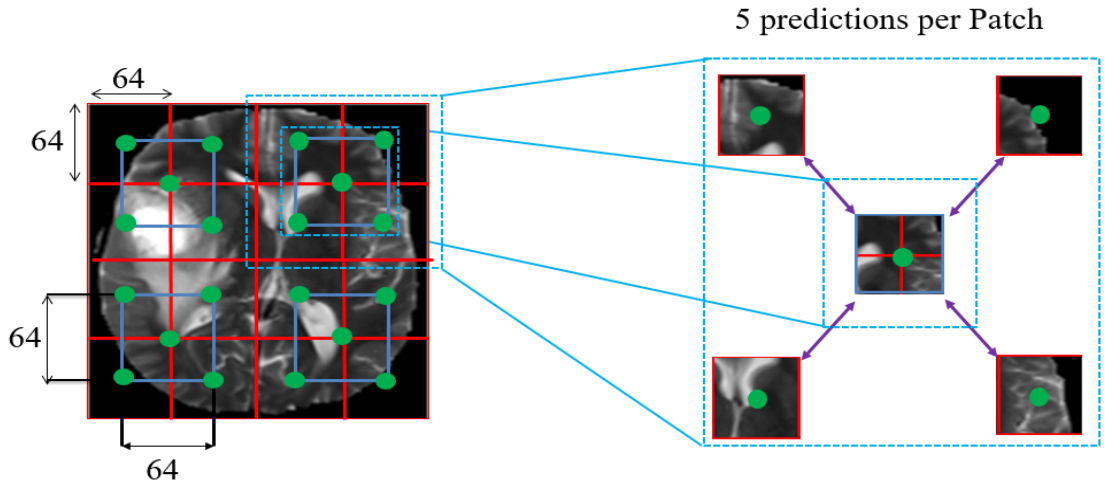
architectures. The application of the second pre-processing step, led to bring the mean value in the range [-1, 1], in other words, the intensities of all regions in addition to healthy and background became between -1 and 1. As we know, the intensity of background pixels of MRI images in BRATS data equals to 0, thus to isolate the zero pixels (i.e. background) from the other regions, we normalized histogram of the MRI images by shifting the zero pixels to another bin outside the range [-1, 1]. We found that the bin -9 in many experiments, gives good results in the training and testing phases.

3.2. Overlapping Patches

Our CNNs architectures are built upon 2D image Patches, in which these architectures predict the pixel's class which is the center of the 2D Patch (i.e. see figure 2, green points). A lot of research (Urban et al., 2014; Pereira et al., 2015; Havaei et al., 2017; Ben naceur et al., 2018) use the technique of Adjacent Patches (see figure 2.a), i.e. using Patches (Patch is a set of pixels) that are next to each other, but the limit of CNNs with Adjacent Patches is that CNNs do not take into account the fact that these Adjacent Patches together make up the entire image (Havaei et al., 2017; Pereira et al., 2017; Zhao et al., 2018). Thus, to overcome this problem and to make the CNNs architectures more accurate, we extract Overlapping Patches (see figure 2.b) which help these architectures to see the local (Red Patches, see figure 2.b) and the larger context (Blue Patches, see figure 2.b). It is well-known that CNNs work by convolving each kernel with the corresponding Patch in the image, then this kernel moves to the next Patch to complete the entire image. Then through the learning process, the CNNs will attempt to classify these adjacent Patches into different classes (i.e., 4 sub-regions), so the final segmented image will be the result of only one pixel prediction (i.e., green point) per Patch as shown in figure 2.a. However, with the Overlapping Patches, the final segmented image will be the result of 5 predictions per Patch as shown in figure 2.b, where the prediction of the other 4 pixels affects the prediction of the first pixel and vice versa for each pixel. Thus with the technique of Overlapping Patches, CNNs could learn to build a relationship among these Patches (Red Patches, see figure 2.a). So, when applying CNNs architecture with the technique of Overlapping Patches, the architectures still able to classify and see the larger context even using small Patches. The explanation of why Overlapping Patches provide more accurate results than Adjacent Patches is due to the prediction of each pixel's label comes from its neighborhoods. Thus, using different Patches around the same pixel, the final decision (i.e. final prediction) of that pixel's label will be based on the vote of the majority of decisions, in other words, the final decision will come from 5 other pre-decisions. In addition, the use of Overlapping Patches provide a data augmentation (Hashemi et al., 2019) of the original BRATS dataset through a better balance of training dataset.



(a) Adjacent Patches.



(b) Overlapping Patches.

Figure 2: Examples of A) Adjacent Patches (one prediction per Patch) and B) Overlapping Patches (5 predictions per Patch) from an input MRI image. The training dataset is created by extracting Overlapping Patches (i.e. Red and blue boxes). Best viewed in color.

3.3. Model and Architectures

MRI provides images that show a contrast between the soft tissue of the brain (or other human organs such as liver) (Akram and Usman, 2011; El-Dahshan et al., 2014), in which these images in many times do not clearly show the border between brain regions, where the brain appears as a single mass. Thus, to extract only the tumoral regions from the whole brain, the proposed model should extract more features

about the healthy tissue and the tumoral regions. To solve this issue, we propose three CNNs architectures that are based on the question of how to maximize the features' representation inside the model, in other words, how to extract much relevant features (Ghebrechristos and Alaghband, 2018) from MRI images.

3.3.1. Visual areas-based interconnected modules

Developing a CNNs architecture based on the rule of using many interconnected modules, is known in the state-of-the-art and it provides a good performance in many applications such as GoogleLetNet model (Szegedy et al., 2015) that is built based on the module of Inception, ResNet model (He et al., 2016) which is based on many modules of residual layers, 3CNet (Ben naceur et al., 2018) is built also on the system of modules. Moreover, the structure of OTP is built on the rule of interconnected areas (i.e. modules): V1, V2, V4, IT (Manassi et al., 2013). Thus, following this rule of interconnected modules, we build our CNNs architectures, where we build first two modules called Sparse Connection OCM (OCM: OCcipito Module) and Dense Connection OCM, these two modules are based on the structure of OTP, this structure is important and critical for object recognition (Desimone et al., 1989) and for the brain's memory (Klüver-Bucy syndrome (KBS), experiments of Klüver and Bucy 1937) (Ono and Nishijo, 1992).

The algorithm of CNNs (LeCun et al., 1989, 1998) originally is inspired by the visual system. In 1962, Hubel and Wiesel (Hubel and Wiesel, 1962) discovered that each type of neurons in the visual system, responds to one specific feature: vertical lines, horizontal lines, shapes ...etc. From this discovery, the algorithm of Neocognitron is developed (Fukushima and Miyake, 1982), but the problem of Neocognitron is: does not have a supervised learning algorithm (LeCun et al., 2015), where in CNNs they solved this problem by using Backpropagation algorithm (Rumelhart et al., 1986). Moreover, we find that the visual system has a specific structure that is composed of several visual areas: V1, V2, V4, IT (Manassi et al., 2013). As we said before, CNNs is inspired by the visual system function, what was the "missing part" of CNNs is the architecture's structure. Thus, in this work, we attempt to inspire from the visual system its internal structure. By inspiring the structure of the visual system, we could add to the function of CNNs a specific architecture.

To determine a shape in the visual system, for example, a person's face, it will be processed from lines and edges at V1 area to shapes at V2 area to objects at V4 area to faces at the end at IT area (Herzog and Clarke, 2014; Manassi et al., 2013). Thus, the OTP structure has four properties (i.e. N:1 to N:4):

1. Retina is the input of the visual system.
2. The visual cortex treats the forms hierarchically; object has different shapes, and shape has different lines and edges and so on.
3. OTP structure is composed of several interconnected visual areas (i.e. V1, V2, V4, IT), where each area responsible for a specific task.
4. Each visual area receives a *Receipted Field* larger than the one at the previous area.

In our work, we attempt to build a CNNs model inspired by the OTP structure, thus, we inspired:

1. The Retina represents the input images.
2. The hierarchical function using convolution operators.
3. We have developed two modules inspired by the area V1 and V2, interconnected to each other using direct and skip connections (see section 3.3.2.1).
4. To take into account the last property (N: 4) of visual cortex, there are two ways:
   (a) Reducing the feature maps at each level using Pooling operation.
   (b) Or using larger kernels at each convolution layer incrementally.

The use of larger kernels (i.e. the property (4.b)) increases the computational time (i.e. increase the number of operations), on the other hand, the use of small kernels, has proved previously that they keep a lot of information and it allows the CNNs model to be deeper in terms of layers (Simonyan and Zisserman, 2014a). Thus, from the above explanations, we will use the first option (i.e. 4.a).

3.3.1.1 Building the first module V1

In this work, we consider the structure of V1 is the same as V2, V4, IT, but the output of these modules is different from module (e.g. V1) to another (e.g. V2). Thus, the module of V1 is composed as shown in figure 3 of: an input (phase 1), then we have convolution layers (phase 2), and a max-pooling layer (phase 3), then another convolution layers (phase 4), upsampling (phase 5) and concatenation layers (phase 6):

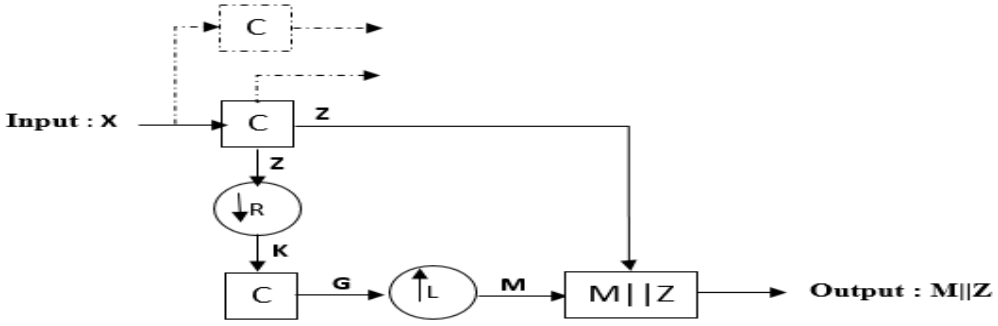


Figure 3: An illustration of visual area V1. C is a convolution operator, Z and G are feature maps, K is the output of Pooling layer, M is a high-resolution feature map, $\downarrow R$ is a pooling layer, $\uparrow L$ is an upsampling operator, $M||Z$ is a concatenation layer between M and Z. Also, the continuous lines are direct and skip connections, and dashed lines are used to build the second module of visual area V1 (i.e. DenseConnectionOCM).

1. First phase: let's consider the Retina as the input (i.e. X) to our model.
2. Second phase: obtaining a hierarchical structure is guaranteed by the method of convolution operation; convolution uses a filter bank (i.e. trainable parameters) to extract many levels of features across the entire input images. Moreover, from our previous work (Ben naceur et al., 2018), we found that

using two consecutive convolution layers with kernel size equals to 3 × 3 provides a high performance. Thus, in this work we will use two consecutive convolution layers Z = Conv(Conv(X)). The output of convolution layer called feature maps Z which are calculated as follows (see equation 1):

$$U = b_i + \sum_{j=1}^{m} W_{ij} * V \tag{1}$$

Where b_i and W_{ij} $(i, j \in \mathbb{N})$ are trainable parameters, called respectively bias and weights. m is the size of the input $(m \in \mathbb{N})$. V and U are respectively input and output, and '*' is the convolution operation. After that we apply a non-linear activation function $Relu(U) = max(0, U)$. In CNNs architecture, W_{ij} is the kernel parameters.

3. Third phase: to reduce the feature maps dimensionality (as described in property N: 4), we use Max-Pooling 2 × 2 operation to provide the minimum reduction of feature maps in addition to the shifting invariant property. Max-Pooling takes the maximum value of each non-overlapping square (i.e. sub-region) of feature map Z. MaxPooling K_{ij} is computed as follows (see equation 2):

$$K_{i,j} = \max_{h} \; Z_{i+h,j+h} \tag{2}$$

Where $h \in \mathbb{N}$ is the size of sub-region, $i, j \in \mathbb{N}$ are the stride values for the vertical and horizontal axes respectively.

4. Fourth phase: using trainable parameters G = Conv(Conv(K)) again to extract more feature maps from the Max-pooling output. Also, to compute the feature maps G, we use equation 1:

5. Fifth phase: to concatenate two types of features maps (i.e. feature maps of phase 2 and phase 5), where these two types of feature maps are not at the same scale. Thus, to obtain higher-resolution feature maps from lower-resolution feature maps in CNNs architecture, there are two techniques: (1) deconvolution, (2) upsampling. The first one uses trainable parameters, but the second technique does not use any trainable parameters, this technique works by inserting zeros padding internally between pixels, where the number of zeros is determined by the stride parameter. The final step is to convolve the sparse feature maps with trainable parameters (i.e. convolution operation) to obtain dense features maps, this technique helps to avoid the problem of Overfitting (Badrinarayanan et al., 2015). The method of upsampling operator works as follows (see equation 3 and 4):

$$\begin{aligned} M_L[n] &= G(n)_{\uparrow L} \\ M_i(n) &= upsampling_{L,n}(G_i) \end{aligned} \tag{3}$$

$$M_i = \begin{cases} G_i(n/L) & , if(n/L) \in \mathbb{N} \\ 0 & , otherwise \end{cases} \tag{4}$$

Upsampling operator works by inserting $L-1$ zeros between $G(n)$ and $G(n+1)$ for all n elements.

6. Sixth phase: is the concatenation layer, where in this layer, we concatenate the feature maps of phase 2 and phase 5.

We used the phases from 1 to 6 to build a Sparse Connection OCM module (see figure 4.a), and a Dense Connection OCM module (see figure 4.b), while in the second module (i.e. Dense Connection OCM module), we have added 1×1 convolution layer (Lin et al., 2013) which operates over N-dimensional volumes and also it's considered as a preliminary classification of the input images. Dense Connection OCM and Sparse Connection OCM modules are two representations of the visual area V1, in which from these two modules, we have deduced three CNNs architectures (see section 3.3.2.2). The motivation of these two modules is they attempt to model the structure of OTP, moreover these modules extract different representations from the input MRI images where this technique is used by GoogleLetNet in its Inception module (Szegedy et al., 2015). We have used two different types of convolution operations (i.e., 3×3 and 1×1) for the module Dense Connection OCM and Sparse Connection OCM to diverse the feature representations of the MRI images:

- Sparse Connection OCM module: applies a strategy of sparse connectivity, where we connect a subset of convolution layers to each other. This module has one input and two outputs (i.e. output 1 and output 2). We refer to this module as SparseConnectionOCM. The details are illustrated in figure 4.a.

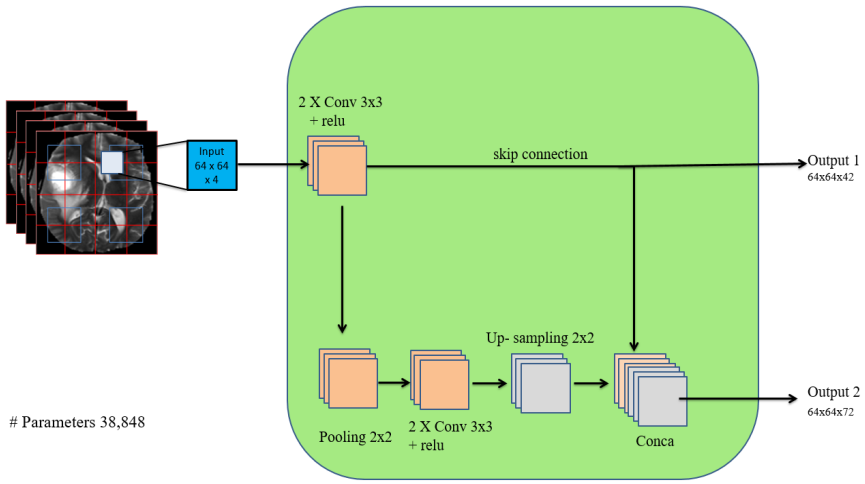
- Dense Connection OCM module: applies two strategies (1) Dense connectivity and (2) 1×1 convolution operation, where we concatenate all high-level feature maps and low-level feature maps in the concatenation layer before the output. This module has one input and one output compared to the previous module (i.e., Sparse Connection OCM). We refer to this module as DenseConnectionOCM. The details are illustrated in figure 4.b.

3.3.2. CNNs architectures

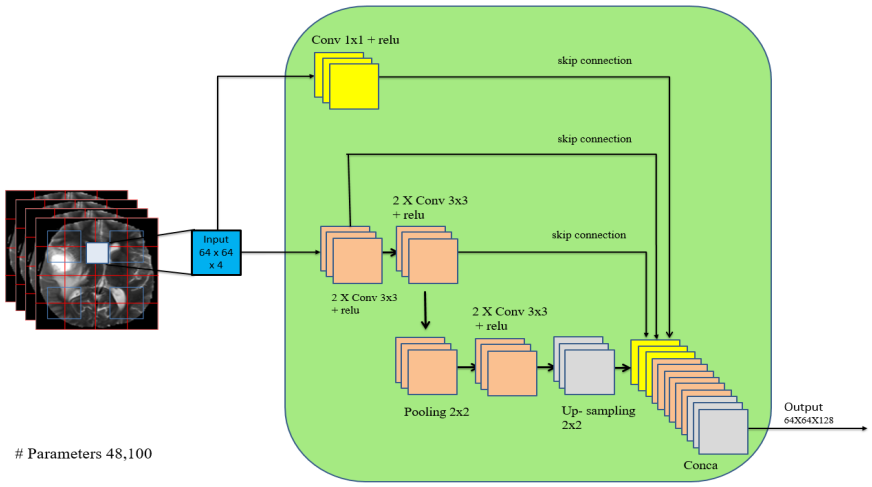
To connect between different modules of CNNs architecture, we use two types of connections: first a direct connection from layer H^{L-1} to layer H^L through a mapping non-linear function f (see equation 5), secondly a special type of connection called skip connection that connect layer H^{L-i} to layer H^L (see section 3.3.2.1).

$$H^L = W^L \times H^{L-1} + b^L \tag{5}$$

Where b^L, W^L are trainable parameters, called respectively bias and weights, after that a non-linear activation function $f(H^L)$ is applied.



(a) SparseConnectionOCM, where we use a skip connection to connect between low-level feature maps and a pre-output concatenation layer. This module has 38,848 parameters.



(b) DenseConnectionOCM, where we use multiple skip connections to connect data and low-level feature maps and pre-output concatenation layer together. This module has 48,100 parameters.

Figure 4: Schematic representation of the SparseConnectionOCM and DenseConnectionOCM. Each orange/purple/yellow box corresponds to multi-channel feature maps. The input represents the multi-modal MRI images. Each output (i.e. output 1, output 2, and output) is an input to the next module. Conv is the convolution operator, 1×1 and 3×3 are the filter's size, pooling 2×2 is the max-pooling operator with 2×2 window. Relu is the non-linear activation function, upsampling is the upsampling operator, Conca is the concatenation layer, skip connection is a connection between two non-successive layers.

3.3.2.1 Skip connections

The use of skip connections is very important; it helps to backpropagate the gradient signal across several layers and leads to a fast convergence (Drozdzal et al., 2016) and since its first introduction (He et al., 2016),

Skip connections have been shown an improvement of accuracy for many computer vision tasks compared to the state-of-the-art methods. For example in the field of biomedical image segmentation (Drozdzal et al., 2016) such as U-net (Ronneberger et al., 2015), and Fully Convolutional Networks (FCN) were applied for semantic segmentation (Long et al., 2015), also for image recognition such as ResNet (He et al., 2016), ResNext (Xie et al., 2017), DenseNet (Huang et al., 2017). The use of skip connections encourage the architecture to reuse the features especially low-level features such as lines, edges; the model uses these low-level features many times with high-level features such as shapes, objects at the same level. Thus, the combination of low-level features and high-level features in MRI images using skip connections, helps to localize the tumor region and then to detect the shape and the boundaries of intra-tumoral structures (i.e. 4 sub-regions). Skip connections connect between two non-successive layers: layer H^{L-i} and layer H^{L} (see equation 6):

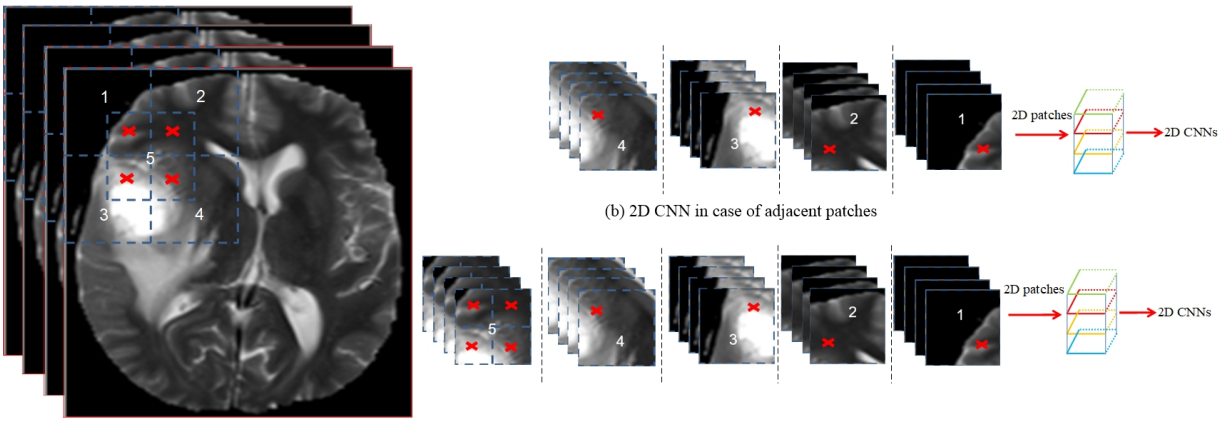
$$H^{L} = W^{L} \times H^{L-i} + b^{L} \tag{6}$$

Where b^{L}, W^{L} are trainable parameters, called respectively bias and weights. H^{L-i}, H^{L} $(i \in \mathbb{N}\ \&\ 2 \leq i)$ are two layers, after that a non-linear activation function $f(H^{L})$ is applied.

3.3.2.2 Sparse and Dense architectures

CNNs are known for its ability to extract many complicated and hierarchical features from images, thus, to develop a deep CNNs architecture, we have either pixel-wise approach or patch-wise approach. In the first approach, CNNs deal with Pixels, while in the second approach, CNNs deal with Patches. Our method in this paper is based on Patch-wise technique, thus CNNs architecture takes as input patches with limited size (in our case 64 x 64), where the size of patches is lower than the size of the full image (240 x 240). The training data is a set of Patches, each 64 x 64. CNNs treat the generated patch, patch-per-patch until the end of pre-defined number of patches, moreover, for each patch, there are three other patches as channels for the multi-sequence of MRI images. The most used technique to generate patches is adjacent patches (see figure 2.a). Then, at prediction time, CNNs predict the segmentation labels separately from others (Havaei et al., 2017); the labels of second patch will be predicted without any relation with the first patch and so on for third, fourth ...etc. In literature, methods such as K-means (Zhang et al., 2019) or conditional random field (Wu et al., 2014), are defined over all pixels or all labels of the full image, like this, the prediction of a label is shared among all pixels, or labels through a proximity distance or a mean-field message respectively. Thus, the patch-wise technique is based on adjacent patches with limited size. From the other hand, to overcome the issue of adjacent patches, we create a link among all these patches of the entire image, where we proposed overlapping patches (see figure 2.b) technique; it works by constructing a new patch by taking 25% from each patch. In this case, overlapping patches provide a data augmentation; the new patch (fifth

patch) is a shifting of each patch (from the remaining four patches) horizontally and vertically on the width axis, then on the height axis respectively (see figure 5). Secondly, we proved through a series of experiments on adjacent patches and overlapping patches, and we found that training CNNs architecture with overlapping patches, provides a good segmentation performance in terms of dice score complete, core, enhancing and average dice compared to adjacent patches (see table 2, and figure 7). Moreover, pixel-wise classification in



(a) Axial MRI image and the patches of the shaded region.

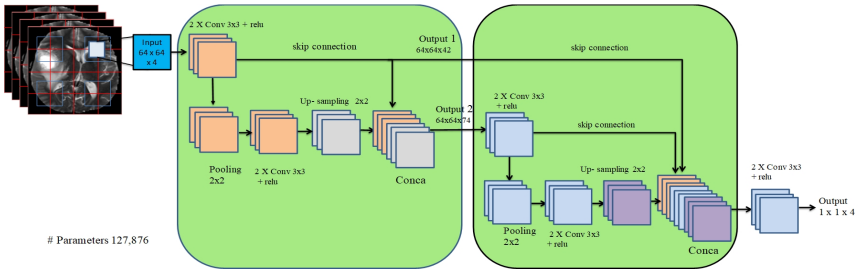
(c) 2D CNN in case of overlapping patches

Figure 5: Examples of (a) a MRI image (Axial view) and the patches of the shaded region, where the red cross signe represents the shared region with the patch number 5. (b) 2D CNNs in case of Adjacent Patches: at each epoch of training we use only 4 patches at the shaded region. (c) 2D CNNs in case of Overlapping Patches: at each epoch we use 4 patch in addition to a new patch (number 5) constructed from the four patches. Best viewed in color.

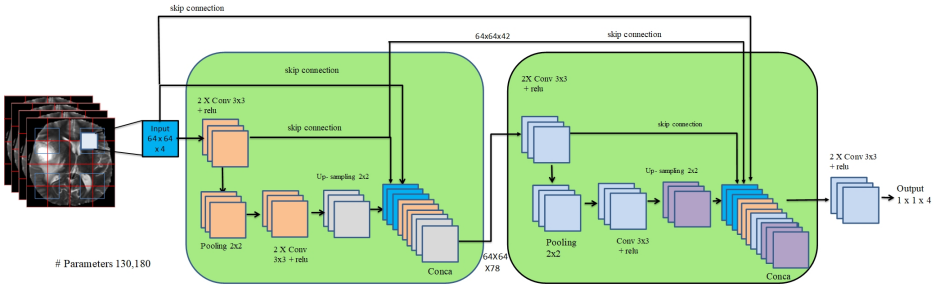
CNNs is computationally expensive such as the architecture of AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014b). From another side, Fully Convolutional Networks (FCN) (Ronneberger et al., 2015; Çiçek et al., 2016; Milletari et al., 2016) have shown a great performance. FCN approach is based on patch-wise classification and it has shown better and more accurate results than pixel-wise classification. In this work we investigate three variants of CNNs architectures:

- Sparse MultiOCM architecture: this architecture made up of two modules of SparseConncentionOCM which has two outputs (i.e. output 1 and output 2), we concatenate the first output (i.e. output 1) with a pre-output concatenation level (i.e. conca layer) of the second module, and the second output (i.e. output 2) is used as an additionnel input to the second module. We refer to this architecture as SparseMultiOCM. The details of this architecture are illustrated in figure 6.a.

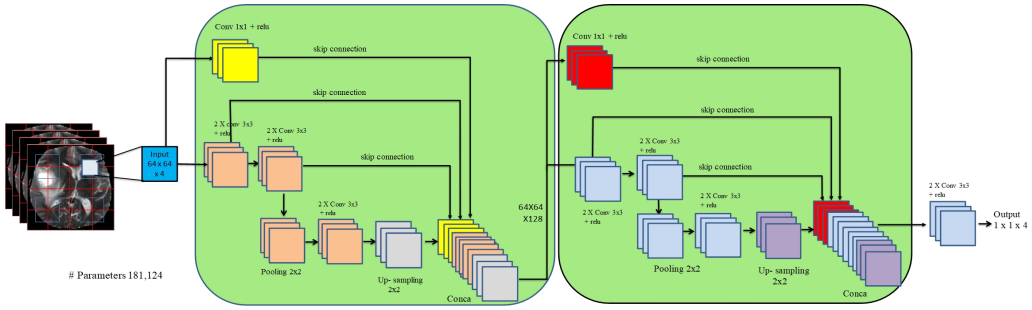
- Input Sparse MultiOCM architecture: this architecture is the same as SparseMultiOCM, where the difference is: in this architecture we investigate the effect of adding directly the input as another concatenation feature maps. We refer to this architecture as InputSparseMultiOCM. The details of this architecture are illustrated in figure 6.b.



(a) SparseMultiOCM, we use two modules of SparseConnectionOCM to build this architecture. This architecture has 127,876 parameters.



(b) InputSparseMultiOCM, we concatenate the input with the pre-output concatenation level in two modules. This architecture has 130,180 parameters.



(c) DenseMultiOCM, we use two modules of DenseConnectionOCM to build the complete architecture. This architecture has 181,124 parameters.

Figure 6: An illustration of the proposed CNNs architectures to solve the problem of fully automatic brain tumor segmentation. Each orange/red/purple/yellow box corresponds to multi-channel feature maps. The output $1 \times 1 \times 4$ is a 1×1 convolution layer with 4 classes of BRATS dataset where we have 4 sub-regions: Healthy tissue, Necrotic and Non-Enhancing tumor, Peritumoral Edema, Enhancing core.

- Dense MultiOCM architecture: we used DenseConncentionOCM module to build this architecture which is another view of OTP, where in this architecture we attempt to extract different feature representations from the input MRI images. We refer to this architecture as DenseMultiOCM. The details of this architecture are illustrated in figure 6.c.

Finally, after building these architectures that are made up of two modules (i.e. SparseConncentionOCM,

DenseConncentionOCM), we have added two consecutive convolution layers before the final output (i.e. output $1 \times 1 \times 4$) to extract unified features. The motivation of adding these consecutive convolution layers is: we have concatenated several feature maps including the input image Patches (i.e. raw pixels) from many levels at the concatenation layer (i.e. conca) of the second module, thus we have heterogeneity of features and raw pixels at one level, these features including raw pixels need to be unified at the same level of representation before classifying them into 4 classes (3 tumoral regions + 1 healthy tissue) using Softmax function.

3.4. Class-Weighting technique

One of the main problems of medical data is unbalanced data, we find, for example, in BRATS dataset, the class of interest (i.e. tumoral region) is the minority class that has almost 2% of data, while the healthy class has 98% of data (Havaei et al., 2017). Training a CNNs model on a such unbalanced dataset will bias toward the majority class (i.e. healthy class), and will produce a low Sensitivity, this problem of unbalanced data is known in medical imaging applications (Hashemi et al., 2019). Thus, to overcome this problem, there are many proposed methods in the state-of-the-art: two training steps (Havaei et al., 2017), median frequency balancing (Badrinarayanan et al., 2015), asymmetric similarity loss function (Hashemi et al., 2019). In our work, we applied two techniques during the training stage (1) extracting Patches from the axial view randomly where each class has the same number of training Patches, this technique is found useful to avoid the problem of unbalanced data (Zhao et al., 2018), (2) also we used class-weighting technique to weight the vote of each class, in other words, this technique gives to each output a specific weight W_j according to its contribution in the segmentation results.

$$p_j = \frac{\exp^{z_j}}{\sum_{k=1}^{K} \exp^{z_k}} \tag{7}$$

$$L(p,q) = -\sum_{j=1}^{K} q_j \times \log(p_j) \tag{8}$$

$$LW(p,q) = -\sum_{j=1}^{K} W_j \times q_j \times \log(p_j) \tag{9}$$

Where K is the number of classes ($K \in \mathbb{N}$), q_j is the j^{th} element of the normalized ground truth vector ($q_j \in \mathbb{R}$) and p_j is the j^{th} element of the estimated vector ($p_j \in \mathbb{R}$) for the class j (j is an integer $\in [1..K]$). W_j ($0 < W_j < 1 \quad and \quad \sum_{j=1}^{K} W_j = 1$) is a specific weight assigned to the class j. L(p,q) and LW(p , q) are two loss functions that represent the error between the estimated vector and the normalized ground truth vector. First formula (i.e. equation 7) called Softmax function, and the second formula (i.e. equation 8) called Cross-entropy Loss function, or log loss, and the third formula (i.e. equation 9) called Weighted

Cross-entropy Loss.

Softmax function gives the estimated vector p_j at the end after each forward propagation of Convolutional Neural Networks, by squashing the outputs to be between 0 and 1, i.e. the outputs become as probabilities. Then these probabilities are fed into Cross-entropy function, but in our case we will use the third function Weighted Cross-entropy Loss, this function after that computes the 'real' error based on the Cross-entropy function and the class-weight W_j for each class. Moreover, to make the prediction of the estimated vector accurate and faster, we use 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 error we need only one operation instead of many operations for all classes. Softmax function attempts to predict this vector (i.e. q vector) after each forward/backward propagation until the end of training. The advantage of Weighted Cross-entropy Loss function is: it could compute the real error for each class, so as a result, it rewards or penalizes only the correct classes. The drawback of this function is how to calculate the weight for each class. To answer this question, we have conducted an experimental analysis of the effect of Class-Weighting technique on the segmentation results (see section 4.4.1).

3.5. Post-processing

To remove some mis-classified and isolated regions (i.e., non-tumor regions) from the segmentation results of our CNNs architectures, we have applied two post-processing techniques:

1. Using a global threshold for each slice (i.e., 2D MRI image) to remove small non-tumoral regions based on connected-components, after many experiments (e.g., 50, 80, 100, 150, 200 ..etc), we found that using 110 pixels provide the best result for this global threshold. Thus, any connected-components smaller than 110 pixels will be removed. We refer to this Post-processing as Post-processing 1.
2. We have tested many morphological operators, in which the opening operation $g(x) = (f \ominus s) \oplus s$, where g(x) is the output and f is the input image, with a structuring element s of size 3 x 3 (see table 1). In this paper, we have used a small structuring element to capture small tumoral regions and to remove them later using an opening operator. Opening operator uses erosion $g(x) = f \ominus s$, then dilation $g(x) = f \oplus s$ in succession, where we noticed that this operator improves the results of the first post-processing (i.e., Post-processing 1). We refer to this Post-processing as Post-processing 2.

We have observed that the application of post-processing 1 + post-processing 2 to the output of the CNNs architectures provides a better segmentation result than the application of a single type of these post-processing techniques.

0	1	0
1	1	1
0	1	0

Table 1: The used structuring element

4. Experimental results

This section presents the results of our experiments and some discussions. Our brain tumor segmentation method is evaluated using BRATS dataset 2018 which has real patient data with Glioblastomas brain tumors (both high- and low-grade). Each patient has 4 types of MRI images, firstly we apply a Pre-processing step (see section 3.1) to these 4 types of MRI images to reduce some noises and to obtain image Patches with a high quality and resolution. Then we extract Overlapping Patches (see section 3.2) from all these types of MRI images to train the CNNs architectures. The last step is Post-processing (see section 3.5), where we apply to the segmentation results of CNNs architectures (i.e., SparseMultiOCM, InputSparseMultiOCM, DenseMultiOCM) two operations: the first one is the global threshold, and the second one is the morphological opening operator. Finally, we explain the libraries and frameworks that are used in our work. Then we evaluate our CNNs architectures with a set of metrics that are used in the state-of-the-art such as Dice score, Sensitivity (Recall), Specificity, and Hausdorff Distance.

4.1. Dataset

We have evaluated our experiments on real patient's data, in which we have used BRATS 2018 dataset. The training set has 210 patient's brain with high-grade (HGG) and 75 patient's brain with low-grade (LGG). Each patient's brain image comes with 4 MRI sequences (i.e., T1, T1c, T2, FLAIR) and the ground truth of 4 segmentation labels which are obtained manually by radiologists experts: Healthy tissue, Necrotic and Non-Enhancing tumor, Peritumoral Edema, Enhancing core. BRATS 2018 validation set contains 66 images of patients with an unknown grade, i.e., the validation set does not have the ground truth labels.

4.2. Implementation

To implement our CNNs architectures, we have used Keras which is a high-level open source Deep Learning library, and Theano (Al-Rfou et al., 2016) as a Back-end, where Theano exploits GPUs to optimize Deep Learning architectures (i.e. to minimize the error). In this work, for computing and comparing the inference time, we used Python environment on Windows 64 bits, Intel Xeon processor CPU @ 3.30 GHz with 8 GB DDR4 RAM, and Nvidia Quadro GPU K2000 with 2 GB GDDR5 memory.

We have tested many techniques for weights initialization: "Glorot_Uniform" (Glorot and Bengio, 2010),

"he_normal" (He et al., 2015) where he_normal initialization provides a fast convergence toward the global minimum compared to Glorot_Uniform, also in this work (He et al., 2016) they found that he_normal initialization gives the best results.

The choice of the optimizer plays a very important role in the learning phase, thus, to obtain the best optimizer for our CNN architectures, we have tested ADAM and RMSProp, Adagrad and SGD optimizers with mini-batch size from 1 to 120, where SGD provides the best results with mini-batch size equals to 8. The initial learning rate (LR) was $LR_0 = 0.001$, then it is deceased by the equation 10:

$$LR_i = 10^{-3} \times 0.99^{LR_{i-1}} \tag{10}$$

Where LR_i ($i \in \mathbb{N}^+$) is the new learning rate, LR_{i-1} is the learning rate of the last epoch, 0.99 is a decreasing factor.

To avoid the Overfitting problem, we have used two methods: (1) BRATS dataset has only 285 MRI images with both high- and low-grade, this number of training data is small to train a Deep Learning model on a multi-classification problem. Thus, we have used a large number of Overlapping Patches (30.660 Patches) that provide a better balance of training data among the four classes (i.e. 4 sub-regions) of Glioblastomas tumors. In addition, the Overlapping Patches technique provides a data augmentation (Hashemi et al., 2019) to the original BRATS dataset. (2) Theoretically, the error on the training and testing sets decreases after many epochs, but in practice, the error on the testing set at a certain point will start to increase again. Thus, at this point which is the minimum, the training will be stopped, this method called early stopping.

All our variants of CNNs architectures are trained from scratch using a large number of MRI image Patches (i.e. $64 \times 64 \times 4$) equals to 30.660 Patches. Where after many experiments (e.g. 80/20, 60/40), we found that splitting dataset into 70% for training (i.e. first phase) 30% testing (i.e. second phase) is the best distribution, then in the validation phase (i.e. third phase), we have 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 have used an online evaluation system [2]. In addition, our CNNs architectures trained using MRI images Patches, but these architectures could segment the 3D MRI volume slice-by-slice.

4.3. Evaluation metrics

To evaluate the performance of our CNNs architectures, we used BRATS online evaluation system, where we upload the segmentation results and the system provides the quantitative evaluations contain Dice score, Sensitivity, Specificity, and Hausdorff distance. For each one of these metrics we have three sub-metrics: Complete (i.e. Necrotic and Non-Enhancing tumor, Peritumoral Edema, Enhancing tumor), Core (i.e.

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

Necrotic and Non-Enhancing tumor, Enhancing tumor), Enhancing (i.e. Enhancing tumor). The evaluation metrics are calculated as follows:

$$\text{Dice (P,T)} = \frac{|P_1 \wedge T_1|}{(|P_1| + |T_1|)/2}, \quad \text{Sensitivity (P,T)} = \frac{|P_1 \wedge T_1|}{|T_1|}, \quad \text{Specificity (P,T)} = \frac{|P_0 \wedge T_0|}{|T_0|},$$

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

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. MRI image is a 3D space (i.e. a set of points in R^3): height, width and depth (i.e. a set of slices), each point on the image is a voxel, and a set of voxels represent a volume. To compute the Hausdorff distance, first we differentiate the voxels of each class (i.e. creating a map) then we save only the voxels that belong to the borders of each class, these voxels represent a surface (*Voronoi surface*). 'p' and 't' are two points on the surface ∂P_1 and ∂T_1 respectively, and d(p,t), d(t,p) are the shortest least-squares distance between point 'p' and 't' and vice versa for d(t,p). Finally, instead of calculating the maximum distance, we compute only 95% of the surface distance between each point from the first surface (i.e. the true lesion region) and the other surface (i.e. the predicted lesion region). This fractional distance is a robust version of Hausdorff because the existence of outliers can have a considerable impact on the last one (i.e. Hausdorff), also, it is proven that it gives better results than the maximum distance (Menze et al., 2015).

4.4. Results

In this section, we evaluate our CNNs architectures on a public BRATS 2018 dataset. We measure the segmentation results using four evaluation metrics (see section 4.3), as well as, using the training and the inference time then we compare our results with the state-of-the-art methods.

4.4.1. The effect of class-weighting on the CNNs architectures

Table 2 shows different experiments that are used to train our CNNs architectures where each line represents a different experiment, also all these experiments are totally independent. We have put our CNNs architectures under a series of different tests and scenarios: (1) with/without class-weighting, (2) with/without Bias, with/without regularization techniques such as (3) Instance normalization, (4) Dropout (with different rates), (5) L2 regularization (with different rates), (6) with/without Nestrov Momentum but these regularization techniques did not improve a lot the segmentation results. In addition, we used Dice score as a metric to evaluate the CNNs architectures, because this metric represents a trade-off between sensitivity (recall) and precision and because it's used in the state-of-the-art BRATS-MICCAI challenges. To demonstrate the performance of Overlapping Patches against Adjacent Patches, we tried different exper-

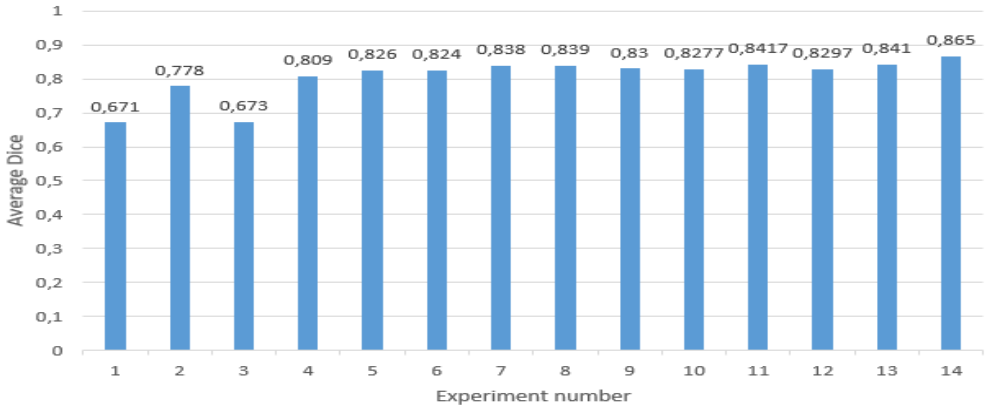


Figure 7: The effect of class-weighting on the CNNs performance. Horizontal axis represents the experiment number in the table 2 (i.e., No.). The vertical axis represents the average Dice score of each experiment (Table 2 column 10).

iments where the initial state (i.e. initial parameters) is the same for both types of Patches:

$$\begin{aligned} Average\ Dice &= \frac{1}{N}\sum_{i} Dice_i \\ &= \frac{Complete + Core + Enhancing}{3} \end{aligned} \tag{11}$$

Where N is the number of Dice score sub-metrics (i.e., N=3).

Table 2: the parameters and the results of different and independent experiments of the class-weighting technique. Healthy, Necrotic and Non-Enhancing tumor (NCR/NET), Peritumoral Edema (Edema), Enhancing tumor represent the different Glioblastomas tumor sub-regions, in which we assigned a percentage of contribution for each class. Moreover, for each experiment we show the Dice score of 10 MRI images from BRATS 2018. The last experiment (No = 14) has the best parameters for our proposed CNNs architectures. Fields with (/) mean we did not apply class-weighting technique.

No.	Patch type	Class-Weights				Dice score			Average Dice
		Healthy	NCR/NET	Edema	Enhancing tumor	Complete	Core	Enhancing	
1	Adjacent	/	/	/	/	0.760	0.595	0.659	0.671
2	Overlap	/	/	/	/	0,902	0,8	0,631	0,778
3	Adjacent	0.3	0.2	0.2	0.3	0.772	0.593	0.655	0.673
4	Overlap	0,3	0,2	0,2	0,3	0,899	0,793	0,736	0,809
5	Overlap	0,3	0,15	0,3	0,25	0,909	0,816	0,752	0,826
6	Overlap	0,3	0,12	0,33	0,25	0,901	0,826	0,745	0,824
7	Overlap	0,3	0,1	0,38	0,22	0,906	0,828	0,779	0,838
8	Overlap	0,3	0,06	0,425	0,215	0,912	0,83	0,775	0,839
9	Overlap	0,29	0,05	0,448	0,212	0,903	0,837	0,7505	0,830
10	Overlap	0,29	0,1	0,4	0,21	0,908	0,828	0,747	0,8277
11	Overlap	0,28	0,08	0,43	0,21	0,91	0,838	0,777	0,8417
12	Overlap	0,27	0,08	0,44	0,21	0,905	0,826	0,758	0,8297
13	Overlap	0,27	0,1	0,43	0,2	0,906	0,842	0,775	0,841
14	Overlap	0,28	0,08	0,43	0,21	0,916	0,866	0,812	0,865

1. First: we used Adjacent Patches (i.e., table 2, first row), and Overlapping Patches (i.e. table 2, second row) without any class-weighting or regularization techniques and as it is shown in the table 2 and figure 7, the first experiment with Adjacent Patches achieved on average (see equation 11) 0.671. However, in the second experiment with Overlapping Patches, the CNNs architectures achieved on average 0.778, which is remarkable that the difference between both experiments is 10.7%.
2. Second: we used Adjacent Patches (i.e., table 2, third row) and Overlapping Patches (i.e., table 2, fourth row), we used also the same mis-classification costs [0.3, 0.2, 0.2, 0.3] for both types of Patches. We have achieved with Adjacent Patches on average (see equation 11) 0.673. However, with Overlapping Patches, the CNNs architectures achieved on average 0.809, the difference between both experiments is 13.6%.

In summary, training CNNs architectures with Overlapping Patches always provide a good segmentation result compared to using adjacent Patches, where we have reported representative results (see table 2) among all the results. Thus, these experiments demonstrate the effectiveness and the efficiency of Overlapping Patches for training CNNs architectures. Moreover, after trying all these experiments with different configurations to improve the segmentation results of CNNs architectures, we found that the last experiment (see table 2, experiment No=14) provides the best mis-classification costs (The class-weight of Healthy tissue=0.28, the class-weight of Necrotic and Non-Enhancing tumor= 0.08, the class-weight of Peritumoral Edema= 0.43, the class-weight of Enhanced tumor= 0.21). Furthermore, we note any adjustment of class-weights doesn't improve the segmentation results. The average Dice score (see equation 11) of each experiment are shown in figure 7.

4.4.2. Segmentation results

In this section, we present the quantitative evaluation on BRATS 2018 validation dataset. First, we evaluate the segmentation results of each CNNs architecture with and without Post-processing 1 and Post-processing 2. Then, we compare these 3 CNNs architectures with the state-of-the-art methods on four evaluation metrics (see section 4.3), and on the training and the inference time in addition to the number of parameters as a metric.

4.4.2.1 Comparison of CNNs architectures with and without post-processing

Table 3, table 4, table 5: show the mean score, standard deviation, median, 25 and 75 quantiles for the four metrics: Dice, Sensitivity, Specificity, and Hausdorff distance. It can be observed that our Deep Learning architectures in these 3 tables achieved high results, in which the Mean Dice scores for the 3 architectures on BRATS validation set are: 0.86, 0.74, 0.74 for whole tumor, tumor core, enhancing tumor respectively. Also, as we can see the Specificity metric is almost 1 (i.e., 100%) which means the two segmentation images (i.e.,

segmentation of radiologist and segmentation of CNNs architectures) are the same for this metric. The Mean Sensitivity metric for the 3 architectures on BRATS validation set is: 0.853, 0.740, 0.774 for whole tumor, tumor core, enhancing tumor respectively. Moreover, applying the first post-processing which is removing a number of connected-components based on a global threshold equals to 110, helped a lot to remove few false positive regions, as we can see the Dice score and the Specificity metrics have been improved after applying this post-processing technique. In addition, after applying Post-processing 2 which is based on a morphological opening operator, we have noticed an improvement in the Hausdorff distance due to the effect of this operator on the class of Enhancing tumor.

Table 3: Evaluation results of SparseMultiOCM architecture with and without Post-processing 1 and Post-processing 2 on BRATS 2018 validation set. WT, TC, ET denote whole tumor, tumor core, enhancing tumor respectively. SparseMultiOCM architecture is trained on Overlapping Patches where the size of these Patches is 64×64×4, and the number of Patches is 30.660.

	Dice score			Sensitivity			Specificity			Hausdorff		
	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET
	SparseMultiOCM architecture											
Mean	0.858	0.746	0.713	0.855	0.756	0.771	0.993	0.996	0.998	21.519	15.160	11.140
Standard deviation	0.111	0.219	0.286	0.154	0.245	0.245	0.005	0.004	0.003	24.263	15.119	23.352
Median	0.891	0.822	0.833	0.898	0.842	0.846	0.995	0.997	0.999	9.306	10.308	2.639
25 quantile	0.860	0.699	0.715	0.856	0.676	0.745	0.990	0.995	0.997	4.031	5.852	1.494
75 quantile	0.915	0.894	0.884	0.937	0.939	0.919	0.997	0.998	0.999	35.176	17.501	6.260
	SparseMultiOCM + post-processing 1											
Mean	0.860	0.748	0.730	0.846	0.756	0.774	0.994	0.996	0.998	8.518	13.466	9.406
Standard deviation	0.125	0.219	0.273	0.168	0.245	0.245	0.005	0.004	0.003	9.546	12.698	21.244
Median	0.897	0.823	0.840	0.888	0.841	0.847	0.996	0.998	0.999	4.243	9.935	2.639
25 quantile	0.868	0.702	0.747	0.846	0.676	0.750	0.992	0.995	0.998	3.0	5.779	1.414
75 quantile	0.923	0.897	0.886	0.935	0.937	0.919	0.998	0.999	0.999	8.944	15.264	5.333
	SparseMultiOCM + post-processing 1 + post-processing 2											
Mean	0.860	0.744	0.725	0.842	0.751	0.753	0.994	0.996	0.998	7.814	13.522	6.713
Standard deviation	0.128	0.220	0.277	0.170	0.243	0.250	0.005	0.004	0.003	8.517	12.656	15.958
Median	0.898	0.812	0.838	0.886	0.835	0.818	0.996	0.998	0.999	4.243	9.648	2.236
25 quantile	0.870	0.697	0.740	0.842	0.686	0.715	0.992	0.995	0.998	3.0	5.679	1.414
75 quantile	0.923	0.897	0.882	0.932	0.934	0.909	0.998	0.999	0.999	8.667	15.516	4.359

These tables (i.e. table 3, table 4, table 5) present also the median values for the validation dataset, SparseMultiOCM, InputSparseMultiOCM, DenseMultiOCM architectures have a high median value which is between 81.2% and 99.9% , these percentages are high, also these results are due to achieving a good and very promising segmentation results for most MRI images, in other words, our CNNs architectures achieved a good segmentation performance for most patients' images, where the median Dice score of our best architecture is 0.902 (whole tumor), 0.845 (tumor core), 0.842 (enhancing core).

According to the segmentation results of 25 quantile: with 25% of results, we are able to detect and segment the Glioblastomas brain tumor in the range 61.9% - 99.8%, as it is observed in the metric of Dice score, Sensitivity, Specificity, and between 94.2% - 98.6% for the Hausdorff distance metric.

According to the segmentation results of 75 quantile: with 75% of results, we are able to detect and segment

Table 4: Evaluation results of InputSparseMultiOCM architecture with and without Post-processing 1 and Post-processing 2 on BRATS 2018 validation set. WT, TC, ET denote whole tumor, tumor core, enhancing tumor respectively. InputSparseMultiOCM architecture is trained on Overlapping Patches where the size of these Patches is 64×64×4, and the number of Patches is 30.660.

	Dice score			Sensitivity			Specificity			Hausdorff		
	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET
	InputSparseMultiOCM architecture											
Mean	0.864	0.746	0.716	0.863	0.755	0.784	0.993	0.996	0.998	16.899	14.669	10.104
Standard deviation	0.121	0.237	0.283	0.163	0.265	0.238	0.006	0.005	0.003	22.072	15.449	19.838
Median	0.899	0.831	0.826	0.909	0.840	0.852	0.995	0.998	0.999	6.117	9.110	3.081
25quantile	0.869	0.678	0.711	0.855	0.680	0.777	0.991	0.996	0.997	3.162	5.171	1.414
75quantile	0.919	0.908	0.881	0.952	0.944	0.930	0.997	0.999	1.0	23.108	19.094	8.062
	InputSparseMultiOCM + post-processing 1											
Mean	0.866	0.747	0.748	0.856	0.753	0.789	0.994	0.996	0.998	10.1868	11.834	6.142
Standard deviation	0.128	0.238	0.256	0.168	0.266	0.238	0.005	0.005	0.003	15.940	9.958	9.286
Median	0.905	0.833	0.839	0.902	0.840	0.854	0.996	0.998	0.999	5.0	8.124	2.236
25quantile	0.873	0.682	0.780	0.846	0.679	0.777	0.992	0.996	0.997	2.871	4.653	1.414
75quantile	0.926	0.909	0.886	0.946	0.944	0.938	0.998	0.999	1.0	8.818	17.117	5.916
	InputSparseMultiOCM + post-processing 1 + post-processing 2											
Mean	0.865	0.744	0.745	0.853	0.748	0.770	0.994	0.996	0.998	10.044	11.504	4.608
Standard deviation	0.130	0.239	0.258	0.170	0.265	0.242	0.005	0.005	0.003	15.833	9.772	6.500
Median	0.904	0.826	0.837	0.900	0.835	0.836	0.996	0.998	0.999	5.0	8.0	2.236
25quantile	0.874	0.682	0.765	0.843	0.677	0.744	0.992	0.996	0.998	3.0	5.0	1.414
75quantile	0.926	0.907	0.886	0.945	0.940	0.933	0.998	0.999	1.0	8.44361	17.117	4.583

Table 5: Evaluation results of DenseMultiOCM architecture with and without Post-processing 1 and Post-processing 2 on BRATS 2018 validation set. WT, TC, ET denote whole tumor, tumor core, enhancing tumor respectively. DenseMultiOCM architecture is trained on Overlapping Patches where the size of these Patches is 64×64×4, and the number of Patches is 30.660.

	Dice score			Sensitivity			Specificity			Hausdorff		
	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET
	DenseMultiOCM architecture											
Mean	0.862	0.737	0.710	0.848	0.710	0.76	0.995	0.998	0.998	14.488	15.673	6.582
Standard deviation	0.139	0.254	0.298	0.173	0.281	0.261	0.005	0.004	0.003	20.215	23.498	11.650
Median	0.902	0.844	0.840	0.889	0.820	0.831	0.996	0.999	0.999	5.337	8.332	2.236
25quantile	0.876	0.668	0.733	0.854	0.625	0.750	0.993	0.997	0.998	3.041	5.099	1.414
75quantile	0.925	0.910	0.878	0.936	0.919	0.922	0.998	0.999	1.0	18.356	14.281	4.359
	DenseMultiOCM + post-processing 1											
Mean	0.861	0.736	0.725	0.841	0.709	0.759	0.995	0.998	0.998	8.514	13.444	5.142
Standard deviation	0.153	0.257	0.289	0.183	0.282	0.263	0.005	0.004	0.003	14.598	21.368	7.608
Median	0.902	0.845	0.842	0.887	0.820	0.835	0.997	0.999	0.999	4.123	8.031	2.236
25quantile	0.875	0.669	0.745	0.845	0.625	0.751	0.993	0.997	0.998	2.871	4.936	1.414
75quantile	0.929	0.911	0.879	0.935	0.919	0.923	0.998	0.999	1.0	7.953	12.083	4.272
	DenseMultiOCM + post-processing 1 + post-processing 2											
Mean	0.860	0.733	0.732	0.838	0.702	0.740	0.995	0.998	0.998	8.521	13.324	4.406
Standard deviation	0.154	0.258	0.282	0.184	0.281	0.269	0.004	0.004	0.003	14.567	21.392	7.070
Median	0.901	0.842	0.842	0.885	0.818	0.816	0.997	0.999	0.999	4.123	8.062	2.236
25quantile	0.874	0.666	0.748	0.842	0.619	0.699	0.993	0.997	0.998	2.980	4.243	1.414
75quantile	0.929	0.913	0.884	0.933	0.908	0.917	0.998	0.999	1.0	7.953	12.114	4.062

the Glioblastomas brain tumor in the range 88.1% - 99.9%, as it is observed in the metric of Dice score, Sensitivity, Specificity, and between 83% - 96% for the Hausdorff distance metric.

In summary, SparseMultiOCM, InputSparseMultiOCM, DenseMultiOCM architectures obtained high segmentation results, and so did with the Post-processing 1 and Post-processing 2, where the Post-processing 2 technique improved a lot the Hausdorff distance metric; Hausdorff distance quantifies how much is the distance between the true position of the pixel in the ground truth and the predicted position in the predicted image, if they are located in the same position, so the distance will become zero, else the distance will become a positive number. The opening operator has the ability to remove small regions and placing them into background. As it is known, Glioblastomas tumors have in most cases one tumoral region, thus using a special designed structuring element will remove some regions that are not captured by first post-processing technique (i.e., connected regions). After applying the morphological opening operator, the first thing we note is a reduction of the number of spreaded false positive regions in the predicted images, also a reduction of the Hausdorff distance. The first conclusion from these segmentation results that Gioblastomas brain tumors contain in most cases (i.e., most patients) one mass (i.e., one connected region) as 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: the class of Enhancing tumor does not have much border with the healthy tissue, and this is demonstrated after applying the Post-processing 2, we have seen a decrease in the surface of misclassified Enhancing tumor region and an improvement in the Hausdorff distance metric.

Finally, these conclusions open a door for more research in the future on how to exploit the spatial features (i.e., the natural structure) of Glioblastomas tumors to improve the segmentation results.

4.4.2.2 Comparison of our CNNs architectures with the state-of-the-art

In this section, we compare the segmentation performance of our 3 variants of CNNs architectures with the state-of-the-art methods.

Table 6: shows a comparison of our CNNs architectures with the state-of-the-art methods, these comparisons are based on four metrics that are described in the section 4.3. It can be observed that there is no single method that surpasses all methods on all metrics simultaneously even to human rater (i.e., Radiologists, first row) and that due to the difficulties and the challenges that we talked about in the introduction (see section 1) and also due to BRATS images that are acquired from several scanners and using different acquisition protocols, thus there is a data heterogeneity problem. The second observation from table 6 that when we applied our CNNs architectures with post-processing techniques, for example, DenseMultiOCM*+ Post-processing 1 & Post-processing 2, we noticed a marked improvement over the results of the architecture itself (i.e., DenseMultiOCM*) in terms of Dice score and Hausdorff distance, thus this improvement demonstrates the

effectiveness of our post-processing techniques. Third observation from table 6 that in overall most methods that are based on CNNs, provide a better segmentation performance compared to machine learning methods such as (Ellwaa et al., 2016) who developed an Iterative [3] method that is based on random forest, however the method of (Ellwaa et al., 2016) provides a result equal to 1 (i.e., 100%) in terms of specificity metric on the class of enhancing tumor (see table 6), this result is better than the results of the remaining methods on the same metric (i.e., specificity of the enhancing tumor class). Also our method InputSparceMultiOCM and the method of (Chang, 2016) have competitive results and that due to the way of building the architectures, where both architectures re-introduce the input (i.e., raw pixels) in the concatenation layer. Therefore, we are not able to completely compare the method of (Chang, 2016) with the rest of metrics (i.e., Specificity, Sensitivity) because the original paper does not mention the segmentation results on these metrics. More-

Table 6: Comparison of the segmentation results of our three proposed models which are noted by adding "*" to the architecture's name with the state-of-the-art methods. WT, TC, ET denote Whole Tumor (complete), Tumor Core, Enhancing Tumor core respectively. Post denotes post-processing. Fields with (-) are not mentioned or given with a specific number in the published work. Numbers in bold are the best values.

Methods	Dice			Specificity			Sensitivity			Hausdorff		
	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET
Human Rater (Menze et al., 2015)	0.88	0.93	0.74	-	-	-	-	-	-	-	-	-
(Havaei et al., 2017)	0.88	0.79	0.73	0.89	0.79	0.68	0.87	0.79	**0.80**	-	-	-
(Chang, 2016)	0.87	**0.81**	0.72	-	-	-	-	-	-	9.1	**10.1**	6.0
Iterative average (HGG + LGG)(Ellwaa et al., 2016)	0.82	0.72	0.56	0.985	0.99	**1.00**	0.835	0.805	0.63	-	-	-
DeepMedic + CRF (Kamnitsas et al., 2017)	0.847	0.67	0.629	-	-	-	0.876	0.607	0.662	-	-	-
(Zhao et al., 2018)	0.84	0.73	0.62	-	-	-	0.82	0.76	0.67	-	-	-
3CNet (Ben naceur et al., 2018)	0.87	**0.81**	**0.83**	0.74	0.78	0.80	**0.89**	**0.84**	0.74	16.86	18,99	13.67
2D-3D model (Mlynarski et al., 2019)	**0.90**	0.808	0.772	-	-	-	-	-	-	-	-	-
SparseMultiOCM*	0.858	0.746	0.713	0.993	0.996	0.998	0.855	0.756	0.771	21.519	15.160	11.140
SparseMultiOCM* + post 1	0.860	0.748	0.730	0.994	0.996	0.998	0.846	0.756	0.774	8.518	13.466	9.406
SparseMultiOCM* + post 1 & 2	0.860	0.744	0.725	0.994	0.996	0.998	0.842	0.751	0.753	**7.814**	13.522	6.713
InputSparseMultiOCM*	0.864	0.746	0.716	0.993	0.996	0.998	0.863	0.7548	0.784	16.899	14.669	10.104
InputSparseMultiOCM* + post 1	0.866	0.747	0.748	0.994	0.996	0.998	0.856	0.753	0.789	10.1868	11.834	6.142
InputSparseMultiOCM* + post 1 & 2	0.865	0.744	0.745	0.994	0.996	0.998	0.853	0.748	0.770	10.044	11.504	4.608
DenseMultiOCM*	0.862	0.737	0.710	**0.995**	**0.998**	0.998	0.848	0.710	0.76	14.488	15.673	6.582
DenseMultiOCM*+ post 1	0.861	0.736	0.725	**0.995**	**0.998**	0.998	0.841	0.709	0.759	8.514	13.444	5.142
DenseMultiOCM*+ post 1 & 2	0.860	0.733	0.732	**0.995**	**0.998**	0.998	0.838	0.702	0.740	8.521	13.324	**4.406**

[3]Iterative method proposes two models: first model for HGG images and the second model for LGG images, where we have averaged the results of these models to compare them with the state-of-the-art methods

over, the method of DeepMedic + CRF (Kamnitsas et al., 2017) provides a good segmentation result for the metric of sensitivity on the whole tumor compared to our CNNs architectures, but for the rest of metrics all our architectures outperformed the method of DeepMedic + CRF (Kamnitsas et al., 2017). Furthermore, DeepMedic + CRF (Kamnitsas et al., 2017), and 2D-3D model (Mlynarski et al., 2019) are a computationally expensive and use a lot of memory due to using 3D MRI images and the need to compute the 3D gradients for each forward/backward pass, in addition, (Kamnitsas et al., 2017) needs other step that is a CRF regularization technique as a post-processing. On the other hand, our method (e.g., DenceMultiOCM) uses only 2D MRI Patches and that makes it efficient in terms of computational time and in terms of memory. The method of (Mlynarski et al., 2019) obtained the best results for the dice metric on the whole tumor class, and these results may be due to using 3D CNNs and using 2D CNNs feature maps as an additional input. Also, we can not do a complete comparison with DeepMedic + CRF (Kamnitsas et al., 2017) and 2D-3D model (Mlynarski et al., 2019) because their original papers do not publish the results of the rest of metrics. (Zhao et al., 2018) developed a CNNs method with an integrated CRF technique in contrast to (Kamnitsas et al., 2017). The method of (Zhao et al., 2018) predicts the final images by averaging the results from 3 views: axial, sagittal, coronal. If we compare this method with DenseMultiOCM*+ Post-processing 1 & Post-processing 2, we find that our CNNs architecture outperformed and provides a better segmentation performance than (Zhao et al., 2018) in terms of all metrics except the sensitivity metric on the tumor core as shown in the table 6.

Finally, our CNNs architectures demonstrate the segmentation performance and the efficiency against the state-of-the-art methods such as (Havaei et al., 2017), 3CNet (Ben naceur et al., 2018), moreover, our methods: InputSparceMultiOCM* + Post-processing 1 and InputSparceMultiOCM* + Post-processing 1 & Post-processing 2 surpassed the human rater (i.e., expert radiologists) for the Dice score metric on the Enhancing tumor class. Furthermore, in this study, besides the problems of fully automatic brain tumor segmentation, we have treated two more issues (i.e., class-imbalance, and the relationship among Patches) which are not addressed in many research such as (Chang, 2016; Ellwaa et al., 2016), where we have solved these two issues using a Class-Weighting technique and Overlapping Patches respectively.

4.4.2.3 Visual segmentation results

In this section, we present examples of visual segmentation of Glioblastomas brain tumors. Figure 8 shows examples of the segmentation results of high-grade Gliomas (HGG) and low-grade Gliomas (LGG) using 3 variants of CNNs architectures (i.e., SparseMultiOCM, InputSparseMultiOCM, DenceMultiOCM). It can be observed that our 3 Deep Learning architectures detect effectively the tumor region (from left to right, column 3 and 8) and its sub-regions: Red: Necrotic and Non-Enhancing tumor, Green: Peritumoral Edema, Yellow: enhancing tumor, also, these architectures are able to detect and segment even small tumoral regions

without any shift of the segmentation results to another spatial position (e.g. patient in row 2, 6, 10) as shown in figure 8, Moreover, we can see that our post-processing techniques improved the spatial segmentation results (from left to right, column 4, 5, 9 and 10).

The drawback of the first post-processing is not a patient-specific (Ben naceur et al., 2018) (i.e., for each patient), but it is very effective when the brain tumor has only one connected region (i.e., one mass). The performance and the precision of the segmentation results in figure 8 due to (1) the Overlapping Patches that introduce more context to the model; using different neighborhood around the same pixels help the CNNs model to take into account several posterior probabilities (i.e., several decisions) which can be maximized over all labels of the training dataset through the supervised learning process, thus the final prediction of the pixel's label of the CNNs model with respect to its parameters will be based on the vote of the majority of decisions, and also due to (2) the technique of *selective attention* that is developed using the rule of interconnected modules, this technique plays a very important role of features maximization.
Finally, after the training and testing phases, it comes the step of deployment and production, where the survived methods from the testing phase will be used in clinics to help the radiologists (see section 4.4.3).

4.4.3. Comparison of Training time and Inference time

Developing a Convolutional Neural Networks model and getting a fast Inference time is one of the challenging issues known in the state-of-the-art. One of the factors that is very important to reduce the Inference time is the number of parameters (i.e., weights) as shown in this paper (Chen et al., 2014), where there is a direct correlation between these two (i.e., the number of parameters and the Inference time). Moreover, SparseMultiOCM architecture has 8 times less parameters compared to 3CNet (Ben naceur et al., 2018) with

Table 7: Results of training time, and inference time and the number of parameters of the state-of-the-art segmentation methods with our 3 proposed models which are noted by adding "*" to the architecture's name. "-" indicates that the information is not provided in the published paper.

Methods	Inference time (seconds)(GPU)	Inference time (seconds)(CPU)	training time (hours)(architecture)	# parameters
(Havaei et al., 2017)	180 (GPU)	-(-)	- (GPU)	802,368
(Chang, 2016)	93 (GPU)	-(-)	2 (GPU)	130,400
Iterative average (HGG + LGG)(Ellwaa et al., 2016)	- (-)	-(-)	- (-)	-
DeepMedic + CRF (Kamnitsas et al., 2017)	30 (GPU)	-(-)	72 (GPU)	-
(Zhao et al., 2018)	540 (GPU)	-(-)	288 (GPU)	-
3CNet (Ben naceur et al., 2018)	19.72 (GPU)	-(-)	5 (GPU)	1,072,420
2D-3D model (Mlynarski et al., 2019)	- (GPU)	-(-)	- (GPU)	-
SparseMultiOCM*	10.77 (GPU)	18.51 (CPU)	1 (GPU)	127,876
InputSparseMultiOCM*	11.68 (GPU)	19.62 (CPU)	1 (GPU)	130,180
DenseMultiOCM*	15.98 (GPU)	24.55 (CPU)	1 (GPU)	181,124

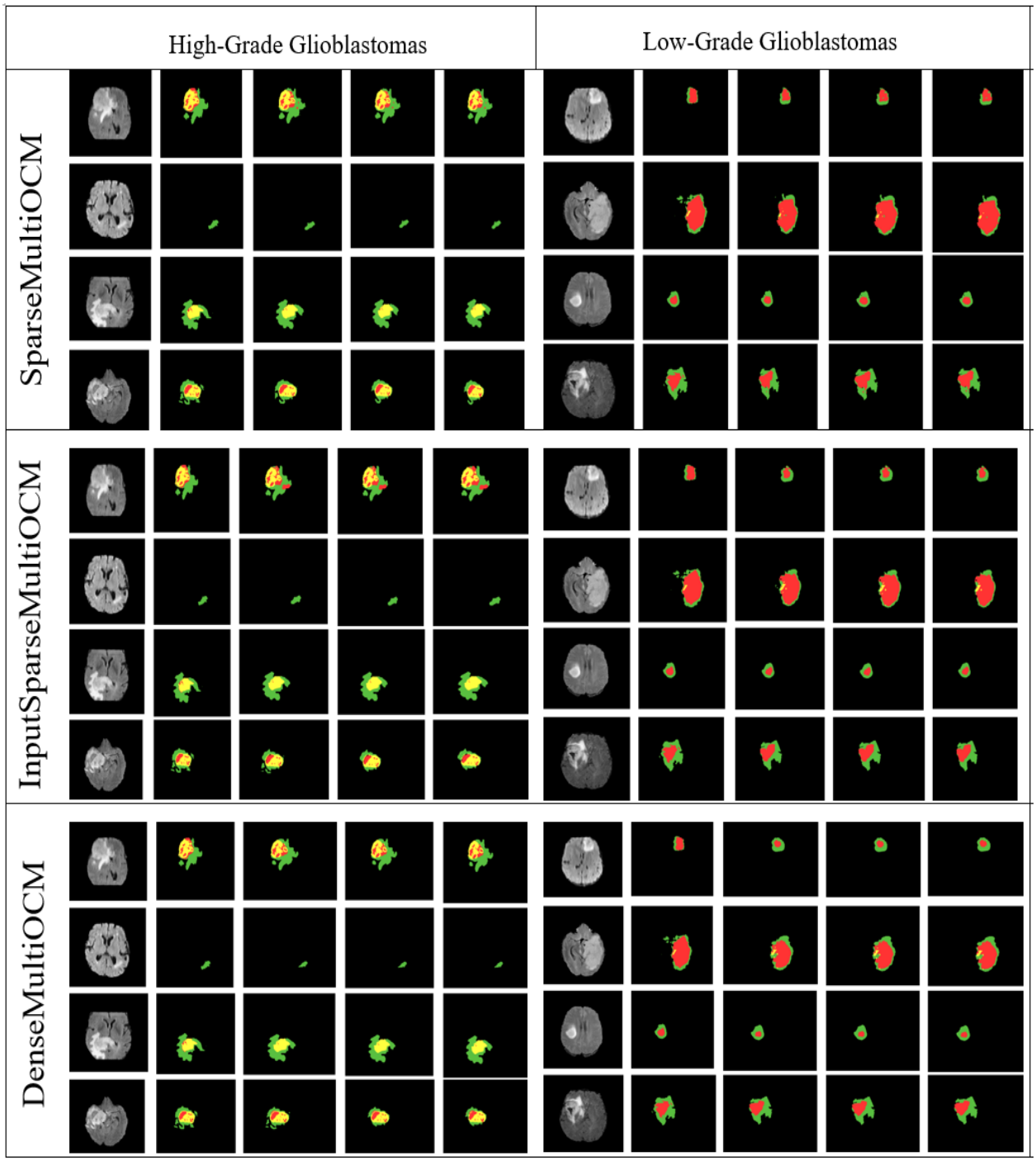


Figure 8: Illustration of three CNNs architectures segmentation results for Glioblastoma brain tumors with High-Grade and Low-Grade Gliomas. Each row represents segmentation results of a different patient's brain tumor from the axial view. From left to right: the first five × twelve matrix are High-Grade Gliomas, and the second five × twelve matrix are Low-Grade Gliomas. Each architecture has 40 different images. The first and six columns are MR scans of different patients (we show Flair MRI modality only for simplicity), the second and the seventh column are the Ground truth segmentation that are created by the radiologists, the third and the eighth columns are the segmentation results of our 3 variants of CNNs architectures, the fourth and ninth column are the segmentation results of each architecture + post-processing 1, the fifth and the tenth column are the segmentation results of each architecture + post-processing 1 +post-processing 2. Colors indicate the tumor regions: Black: Healthy and background, Red: Necrotic and Non-Enhancing tumor, Green: Peritumoral Edema, Yellow: Enhancing tumor. Best viewed in color

the same GPU, and we have noticed that the Inference time decreased to almost the half: 10.77 seconds where in (Ben naceur et al., 2018) is 19.72 seconds, in addition, the inference time on CPU decreased to 18.51 seconds, almost 7% faster. Also, the use of Overlapping Patches helps to reduce the Inference time due to using small Patches; it is known that CNNs model is a set of feature maps that represent the extracted features from the input data, and the use of small input Patches leads to obtain small feature maps, thus getting small feature maps size means the reduce of the required computational time. Moreover, in (Szegedy et al., 2017) have shown that the use of skip connections helps to reduce the training time.

Table 7 shows a comparison of our three Deep Learning architectures with the state-of-the-art methods, this comparison is based on 3 metrics: inference time, training time, and the number of parameters. Our SparseMultiOCM architecture is the fastest method among all other methods that are described in the table 7, SparseMultiOCM could segment the whole brain tumor only in 10.77 seconds on GPU and 18.51 seconds on CPU, where the radiologist spends on average 4 hours (Kaus et al., 2001) to segment only one brain tumor. However, SparseMultiOCM could segment 1415 brain tumor (i.e., 1415 patients) in the same period of time (i.e., 4 hours). Also, it can be observed that our architectures (i.e., SparseMultiOCM, InputSparse-MultiOCM, DenceMultiOCM) outperformed all methods that are cited in the table 7 in terms of inference time (10.77, 11.68, 15.98 on GPU and 18.51, 19.62, 24.55 on CPU respectively) and the needed time for training (i.e., one hour of training on GPU) with a lower number of parameters.

SparseMultiOCM model decreased a lot the inference time, almost 16 times faster on GPU and 9 times faster on CPU in comparison with the state-of-the-art methods such as (Havaei et al., 2017), and almost 50 times faster on GPU and 29 times faster on CPU than the method of (Zhao et al., 2018). After getting the best class-weights (see section 4.4.1), we needed almost one hour of training on GPU to get this segmentation performance in comparison with (Zhao et al., 2018), their CNNs model took 12 days (i.e., 288 hours) of training.

As a summary, the design of Deep CNNs architectures is very important, where as shown in Table 7 our proposed architectures (i.e., SparseMultiOCM, InputSparseMultiOCM, DenseMultiOCM) achieve the tradeoff between the segmentation performance and the inference time. Moreover, the inference time on average 12 seconds on GPU and 20 seconds on CPU, i.e., the number of operations and parameters of our proposed architectures are optimized to be executed on GPU and CPU in a small range of 14 seconds compared to the state-of-the-art methods. The proposed design allows the architectures to reduce the required time for the segmentation of the MRI images; our architectures design technique is different from the other works (Chang, 2016; Ellwaa et al., 2016; Havaei et al., 2017; Kamnitsas et al., 2017; Zhao et al., 2018; Ben naceur et al., 2018; Mlynarski et al., 2019), where they proposed computationally expensive architectures even on GPU with a large memory. The advantage of our architectures is due to the features maximization through selective attention technique, Overlapping patches technique and the rule of interconnected modules.

In the following section, we explain more advantages and computational benefits of our proposed architectures with more details.

4.4.3.1 Computational benefits

In this paper, we reduced the training time in addition to the inference time (test time), to almost one hour for the training time, and for the inference time on average 12 seconds on GPU and 20 seconds on CPU, these computational benefits are due to: (1) an inspired architecture based on the selective attention technique to maximize the feature representations and to reduce features redundancy, (2) we use two consecutive layers of convolution, where this technique has been proven that it gives a good performance in terms of segmentation and speed (Ben naceur et al., 2018), (3) limited number of filters in each layer, as the use of many filters at each layer may lead to redundant and duplicate features in the architecture, (4) also the use of fully convolutional networks (Long et al., 2015) instead of fully connected layers as the classifier, helped the architecture to give accurate results and to reduce the training time due to sparse connectivity, (5) the use of skip connections helps to reduce the time of training (Szegedy et al., 2017), (6) applying convolution to multi-resolution features after applying Max-pooling and Up-sampling allows us to detect relevant features with lower parameters and operations, in addition, this technique contributed to reduce the inference time (7) and finally, the use of small filters (3 x 3) have proved in (Simonyan and Zisserman, 2014b) that they keep a lot of information and reduce the number of multiplications between the input (or the previous layer) and the large number of filters. Moreover, the metrics of inference time and training time in addition to the number of parameters are very important when it comes to production and deployment of these methods in real practice clinics and hospitals with real MRI images. To focus on the effectiveness of Deep learning models, we measured the metric of Dice whole density, i.e., Dice whole divided by the number of parameters. The best model is the one that has a high Dice whole density, in other words, the model that uses effectively its parameters. Measuring information density is one of the most used metrics to measure the effectiveness of deep learning models; information density shows us the ability of each model to use its parameters efficiently (Canziani et al., 2016), (Bianco et al., 2018). In figure 9.a, we can clearly see that the most efficient models among the state-of-the-art are SparseMultiOCM, (Chang, 2016) and InputSparseMultiOCM, where their Dice whole density is greater than 6.66. Also, from the figure 9.b of density information, it can be seen that although 3CNet (Ben naceur et al., 2018) and (Havaei et al., 2017) have a large number of parameters but they do not take the advantage of their capacity in the learning process; these models have a huge number of parameters that do not contribute to the segmentation results. Moreover, our proposed SparseMultiOCM model is efficiently designed for the given task (brain tumor segmentation problem), it achieved the highest Dice whole density score among the state-of-the-art methods with 9x less parameters.

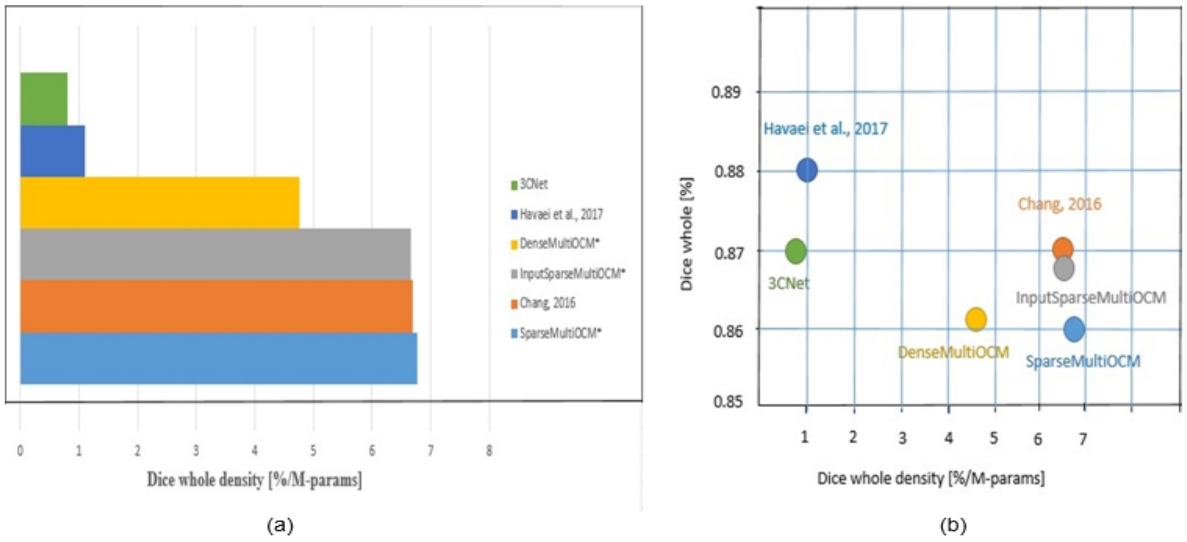


Figure 9: Dice whole density (a) and Dice whole vs. Dice whole density (b). Dice whole Density metric gives us an insight about the efficiency of each model. Missing models are due to the lack of parameter numbers in their published papers. Best viewed in color.

5. Discussions and conclusion

In this paper, we presented a fully automatic brain tumor segmentation method using End-to-End Deep Convolutional Neural Networks. To build a Deep neural network architecture more robust and based on an inspiration far from the limited choices (features maps, kernels, strides, activation functions, connectivity between layers), we proposed a Deep CNNs architecture inspired from selective attention technique of the Occipito-temporal pathway. This inspiration allowed to develop a *selective attention* model in three variants of CNNs architectures (i.e. SparseMultiOCM, InputSparseMultiOCM, DenseMultiOCM), where these architectures demonstrated their segmentation performances in Multi-modal MRI images of low- and high-grade Glioma patients and through BRATS online evaluation system.

The first step of our brain tumor segmentation pipeline is to a find a pre-processing method to enhance the quality of MRI images, where we proposed three steps of pre-processing, then we demonstrated that these steps improve the segmentation results. After that, we post-processed the segmentation results of CNNs architectures by two steps: removing small connected-components (regions) less than 110 pixels, then, we proposed a morphological opening operator with a structuring element showing its performance in figure 8 and tables 3 - 5.

Our three variants of CNNs architectures are built upon 2D Axial images Patches, where these Patches are equally and randomly sampled from the training dataset, i.e. each class has the same number of Patches. Moreover, using 2D CNNs helped to solve partially the issue of long rang context. Also, this technique of generating these image Patches in addition to using a Weighted Cross-entropy loss function during the

training helped to avoid the class-imbalance problem. Moreover, to obtain the correct number of distribution for each class, we conducted an experimental analysis of the effect of Class-Weighting on the segmentation results.

We also proposed a new Overlapping Patches technique, this technique helps our three variants CNNs architectures to build a relationship among the image Patches which led these architectures to take into account the larger context without using larger Patches. Also, through the table 2, we have demonstrated that using Overlapping Patches is better than using Adjacent Patches to train CNNs architectures.

Our proposed CNNs architectures are based on the rule of interconnected modules which are connected by skip connections to build the technique of *selective attention* that plays a very important role of maximizing the features representation inside the architectures. Where short skip connections inside the same module help to complete the missing information during pooling layers and convolution striding and long skip connections between two modules encourage the features reuse which help the architectures to combine low-level and high-level features and to better locate the boundaries of the tumor regions.

The advantages of our proposed CNNs architectures (i) first the segmentation results of these architectures on the four metrics (i.e. Dice, Sensitivity, Specificity, Hausdorff distance) are similar to radiologist experts, also our method InputSparceMultiOCM* + Post-processing 1 surpassed the human rater (i.e. radiologist) for the Mean Dice score metric on the Enhancing tumor class. (ii) The efficient design of our proposed CNNs architectures in addition to slice-by-slice segmentation technique, makes these three variants not only computationally efficient but also these architectures segment the whole brain and save the patient's life on average 12 seconds on GPU and 20 seconds on CPU, on the other hand, the segmentation mission takes from the radiologist on average 4 hours to obtain the segmentation results. (iii) These architectures are an End-to-End Deep Leaning approach, and fully automatic without any user interventions. (iv) DenseMultiOCM architecture with Post-processing 1 & Post-processing 2 is the best architecture in terms of segmentation results, while SparseMultiOCM achieved the tradeoff between the number of parameters and the dice whole score, this tradeoff is measured by dice whole density score. Moreover, these two CNNs architectures are suitable for adopting in research and as a part of different clinical settings.

Finally, the obtained results illustrate that our three variants architectures achieved the state-of-the-art performance on both accuracy and speed, and give a very accurate result for the segmentation of Glioblastomas tumors, also these architectures detect effectively the tumor regions and its subregions even small regions as shown in figure 8. In addition, our experimental results show that the proposed Deep Learning architectures improved the segmentation results and this improvement is demonstrated through a separated validation dataset with real MRI images.

As a perspective of this research, we intend to investigate the 3D Overlapping Patches of Axial, Sagittal and Coronal views to further improve the tumor segmentation performance. Moreover, to solve the prob-

lem of CNN performance degradation during the training on heterogeneous data such as BRATS dataset, our future study is to build a new type of architecture to generate a synthetic data to take into account different clinical scenarios. The implementation of our method will be made publicly available online on https://github.com/MostefaBen.

References

Abd-Ellah, M. K., Awad, A. I., Khalaf, A. A., and Hamed, H. F. (2016). Design and implementation of a computer-aided diagnosis system for brain tumor classification. In *Microelectronics (ICM), 2016 28th International Conference on*, pages 73–76. IEEE.

Akram, M. U. and Usman, A. (2011). Computer aided system for brain tumor detection and segmentation. In *Computer Networks and Information Technology (ICCNIT), 2011 International Conference on*, pages 299–302. IEEE.

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., et al. (2016). Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint*.

Axel, D., Mohammad, H., David, W.-F., Antoine, B., Lam, T., Pierre-Marc, J., Aaron, C., Hugo, L., Chris, P., and Yoshua, B. (2014). Brain tumor segmentation with deep neural networks. *Proceedings MICCAI-BRATS*, pages 01–05.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.

Ben naceur, M., Kachouri, R., Akil, M., and Saouli, R. (2019). A new online class-weighting approach with deep neural networks for image segmentation of highly unbalanced glioblastoma tumors. In *International Work-Conference on Artificial Neural Networks*, pages 555–567. Springer.

Ben naceur, M., Saouli, R., Akil, M., and Kachouri, R. (2018). 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.

Bhandarkar, S. M. and Nammalwar, P. (2001). Segmentation of multispectral mr images using a hierarchical self-organizing map. In *Computer-Based Medical Systems, 2001. CBMS 2001. Proceedings. 14th IEEE Symposium on*, pages 294–299. IEEE.

Bianco, S., Cadene, R., Celona, L., and Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277.

Canziani, A., Paszke, A., and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

Caselles, V., Catté, F., Coll, T., and Dibos, F. (1993). A geometric model for active contours in image processing. *Numerische mathematik*, 66(1):1–31.

Cates, J. E., Whitaker, R. T., and Jones, G. M. (2005). Case study: an evaluation of user-assisted hierarchical watershed segmentation. *Medical Image Analysis*, 9(6):566–578.

Chang, P. D. (2016). Fully convolutional neural networks with hyperlocal features for brain tumor segmentation. In *Proceedings MICCAI-BRATS Workshop*, pages 4–9.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer.

Clark, M. C., Hall, L. O., Goldgof, D. B., Velthuizen, R., Murtagh, F. R., and Silbiger, M. S. (1998). Automatic tumor segmentation using knowledge-based techniques. *IEEE transactions on medical imaging*, 17(2):187–201.

Desimone, R., Moran, J., and Spitzer, H. (1989). Neural mechanisms of attention in extrastriate cortex of monkeys. In *Dynamic interactions in neural networks: Models and data*, pages 169–182. Springer.

Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., and Pal, C. (2016). The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, volume 10008, pages 179–187. Springer.

El-Dahshan, E.-S. A., Mohsen, H. M., Revett, K., and Salem, A.-B. M. (2014). Computer-aided diagnosis of human brain tumor through mri: A survey and a new algorithm. *Expert systems with Applications*, 41(11):5526–5545.

Ellwaa, A., Hussein, A., AlNaggar, E., Zidan, M., Zaki, M., Ismail, M. A., and Ghanem, N. M. (2016). Brain tumor segmantation using random forest trained on iteratively selected patients. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 129–137. Springer.

Fletcher-Heath, L. M., Hall, L. O., Goldgof, D. B., and Murtagh, F. R. (2001). Automatic segmentation of non-enhancing brain tumors in magnetic resonance images. *Artificial intelligence in medicine*, 21(1-3):43–63.

Fukushima, K. and Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern recognition*, 15(6):455–469.

Ghebrechristos, H. and Alaghband, G. (2018). Exploring deep learning using information theory tools and patch ordering. *International Conference on Learning Representations*.

Gibbs, P., Buckley, D. L., Blackband, S. J., and Horsman, A. (1996). Tumour volume determination from mr images by morphological segmentation. *Physics in Medicine & Biology*, 41(11):2437.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Goetz, M., Weber, C., Binczyk, F., Polanska, J., Tarnawski, R., Bobek-Billewicz, B., Koethe, U., Kleesiek, J., Stieltjes, B., and Maier-Hein, K. H. (2016). Dalsa: domain adaptation for supervised learning from sparsely annotated mr images. *IEEE transactions on medical imaging*, 35(1):184–196.

Goodman, C. C. and Fuller, K. S. (2014). *Pathology-E-Book: Implications for the Physical Therapist*. Elsevier Health Sciences.

Gupta, A. and Dwivedi, T. (2017). A simplified overview of world health organization classification update of central nervous system tumors 2016. *Journal of neurosciences in rural practice*, 8(4):629.

Hashemi, S. R., Salehi, S. S. M., Erdogmus, D., Prabhu, S. P., Warfield, S. K., and Gholipour, A. (2018). 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.

Hashemi, S. R., Salehi, S. S. M., Erdogmus, D., Prabhu, S. P., Warfield, S. K., and Gholipour, A. (2019). Asymmetric similarity loss function to balance precision and recall in highly unbalanced deep medical image segmentation. *IEEE Access*, 7:1721–1735.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., and Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. pages 770–778.

Herzog, M. H. and Clarke, A. M. (2014). Why vision is not both hierarchical and feedforward. *Frontiers in computational neuroscience*, 8:135.

Holland, E. C. (2001). Progenitor cells and glioma formation. *Current opinion in neurology*, 14(6):683–688.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR*, volume 1, page 3.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Işın, A., Direkoğlu, C., and Şah, M. (2016). Review of mri-based brain tumor image segmentation using deep learning methods. *Procedia Computer Science*, 102:317–324.

Kamnitsas, K., Ferrante, E., Parisot, S., Ledig, C., Nori, A. V., Criminisi, A., Rueckert, D., and Glocker, B. (2016). Deepmedic for brain tumor segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 138–149. Springer.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. (2017). Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78.

Kaus, M. R., Warfield, S. K., Nabavi, A., Black, P. M., Jolesz, F. A., and Kikinis, R. (2001). Automated segmentation of mr images of brain tumors. *Radiology*, 218(2):586–591.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323.

Lefohn, A. E., Cates, J. E., and Whitaker, R. T. (2003). Interactive, gpu-based level sets for 3d segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 564–572. Springer.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

Liu, L., Zhang, H., Rekik, I., Chen, X., Wang, Q., and Shen, D. (2016). Outcome prediction for patient with high-grade gliomas from brain functional and structural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 26–34. Springer.

Logeswari, T. and Karnan, M. (2009). An improved implementation of brain tumor detection using segmentation based on soft computing. *Journal of Cancer Research and Experimental Oncology*, 2(1):006–014.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation . *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

Louis, D. N., Perry, A., Reifenberger, G., Von Deimling, A., Figarella-Branger, D., Cavenee, W. K., Ohgaki, H., Wiestler, O. D., Kleihues, P., and Ellison, D. W. (2016). The 2016 world health organization classification of tumors of the central nervous system: a summary. *Acta neuropathologica*, 131(6):803–820.

Manassi, M., Sayim, B., and Herzog, M. H. (2013). When crowding of crowding leads to uncrowding. *Journal of Vision*, 13(13):10–10.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., and Burren, e. (2015). The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024.

Menze, B. H., Van Leemput, K., Lashkari, D., Weber, M.-A., Ayache, N., and Golland, P. (2010). A generative model for brain tumor segmentation in multi-modal images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 151–159. Springer.

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE.

Mlynarski, P., Delingette, H., Criminisi, A., and Ayache, N. (2019). 3d convolutional neural networks for tumor segmentation using long-range 2d context. *Computerized Medical Imaging and Graphics*, 73:60–72.

Ono, T. and Nishijo, H. (1992). Neurophysiological basis of the klüver-bucy syndrome: Responses of monkey amygdaloid neurons to biologically significant objects.

Ostrom, Q. T., Bauchet, L., Davis, F. G., Deltour, I., Fisher, J. L., Langer, C. E., Pekmezci, M., Schwartzbaum, J. A., Turner, M. C., Walsh, K. M., et al. (2014). The epidemiology of glioma in adults: a "state of the science" review. *Neuro-oncology*, 16(7):896–913.

Pereira, S., Oliveira, A., Alves, V., and Silva, C. A. (2017). On hierarchical brain tumor segmentation in mri using convolutional neural networks: a preliminary study. In *Bioengineering (ENBENG), 2017 IEEE 5th Portuguese Meeting on*, pages 1–4. IEEE.

Pereira, S., Pinto, A., Alves, V., and Silva, C. A. (2015). Deep convolutional neural networks for the segmentation of gliomas in multi-sequence mri. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 131–143. Springer.

Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., and Dean, J. (2018). Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*.

Prastawa, M., Bullitt, E., Moon, N., Van Leemput, K., and Gerig, G. (2003). Automatic brain tumor segmentation by subject specific modification of atlas priors1. *Academic radiology*, 10(12):1341–1348.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.

Schneider, T., Mawrin, C., Scherlach, C., Skalej, M., and Firsching, R. (2010). Gliomas in adults. *Deutsches Ärzteblatt International*, 107(45):799.

Simonyan, K. and Zisserman, A. (2014a). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, A., Bajpai, S., Karanam, S., Choubey, A., and Raviteja, T. (2012). Malignant brain tumor detection. *International Journal of Computer Theory and Engineering*, 4(6):1002.

Stadlbauer, A., Moser, E., Gruber, S., Buslei, R., Nimsky, C., Fahlbusch, R., and Ganslandt, O. (2004). Improved delineation of brain tumors: an automated method for segmentation based on pathologic changes of 1h-mrsi metabolites in gliomas. *Neuroimage*, 23(2):454–461.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:1–9.

Tustison, N., W. M. A. B. (2013). Ants and árboles. In *Proceedings MICCAI-BRATS challenge*, pages 47–50.

Urban, G., Bendszus, M., Hamprecht, F., and Kleesiek, J. (2014). Multi-modal brain tumor segmentation using deep convolutional neural networks. *MICCAI BraTS (Brain Tumor Segmentation) Challenge. Proceedings, winning contribution*, pages 31–35.

Wu, W., Chen, A. Y., Zhao, L., and Corso, J. J. (2014). Brain tumor detection and segmentation in a crf (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features. *International journal of computer assisted radiology and surgery*, 9(2):241–253.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE.

Yang, S. and Ramanan, D. (2015). Multi-scale recognition with dag-cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1215–1223.

Young, R. J. and Knopp, E. A. (2006). Brain mri: tumor evaluation. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 24(4):709–724.

Zhang, C., Shen, X., Cheng, H., and Qian, Q. (2019). Brain tumor segmentation based on hybrid clustering and morphological operations. *International Journal of Biomedical Imaging*, 2019.

Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57.

Zhao, X., Wu, Y., Song, G., Li, Z., Zhang, Y., and Fan, Y. (2018). A deep learning model integrating fcnns and crfs for brain tumor segmentation. *Medical image analysis*, 43:98–111.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.