Semantic Segmentation of Volumetric Medical Images with 3D Convolutional Neural Networks

Alejandra Márquez Herrera
Universidad Católica San Pablo
Arequipa, Peru
alejandra.marquez@ucsp.edu.pe

and

Alex Jesus Cuadros-Vargas
Universidad Católica San Pablo
Arequipa, Peru
alex@ucsp.pe

and

Helio Pedrini
University of Campinas
Campinas, Brazil
helio@ic.unicamp.br

Abstract

A neural network is a mathematical model that is able to perform a task automatically or semi-automatically after learning the human knowledge that we provided. Moreover, a Convolutional Neural Network (CNN) is a type of neural network that has shown to efficiently learn tasks related to the area of image analysis, such as image segmentation, whose main purpose is to find regions or separable objects within an image. A more specific type of segmentation, called semantic segmentation, guarantees that each region has a semantic meaning by giving it a label or class. Since CNNs can automate the task of image semantic segmentation, they have been very useful for the medical area, applying them to the segmentation of organs or abnormalities (tumors). This work aims to improve the task of binary semantic segmentation of volumetric medical images acquired by Magnetic Resonance Imaging (MRI) using a pre-existing Three-Dimensional Convolutional Neural Network (3D CNN) architecture. We propose a formulation of a loss function for training this 3D CNN, for improving pixel-wise segmentation results. This loss function is formulated based on the idea of adapting a similarity coefficient, used for measuring the spatial overlap between the prediction and ground truth, and then using it to train the network. As contribution, the developed approach achieved good performance in a context where the pixel classes are imbalanced. We show how the choice of the loss function for training can affect the final quality of the segmentation. We validate our proposal over two medical image semantic segmentation datasets and show comparisons in performance between the proposed loss function and other pre-existing loss functions used for binary semantic segmentation.

Keywords: Semantic Segmentation, Medical Images, Convolutional Neural Network, Loss Function, Class Imbalance

1 Introduction

Semantic segmentation in images is a type of segmentation that aims to find regions within an image that has a semantic meaning. In other words, it ensures that pixels grouped into a particular region belong to the same class [1–6]. The task of semantic segmentation involves performing classification over every pixel of the image. In a binary segmentation, a pixel is classified either belonging to the foreground or to the background.

A Convolutional Neural Network (CNN), is a type of neural network that can efficiently perform semantic segmentation in images. That is why, in the medical field, it has been used to automatically segment organs or abnormalities in 2D and 3D Magnetic Resonance Imaging (MRI). CNNs are very useful in this scenario, given that a single MRI image can contain tens to hundreds of images, and it is a slow and difficult task for a human to do it manually.

Loss functions are a fundamental part of training neural networks since they are responsible for measuring the degree of error between predictions and their ground truth and, based on that metric, network parameters are refreshed for improving prediction.

A typical classification loss function, such as cross-entropy, may not always be efficient in a semantic segmentation context, especially when the number of pixels belonging to different classes is imbalanced, which occurs frequently on medical image datasets. If we segmented an organ occupying only 5% of the total number of pixels in an image and we failed to classify all the pixels belonging to the organ as belonging to the background instead, the cross-entropy loss function would measure our prediction as being 95% accurate. It is clear that, in this scenario, the metric was inappropriate since our prediction completely failed to classify the pixels belonging to the object. That is why some other loss function alternatives for semantic segmentation emerged for better guiding the training process.

The main purpose of this work is to develop and analyze a loss function for improving the semantic binary segmentation of volumetric medical images, using the Dense V-Net architecture [7] as base model, and applying the idea of adapting a similarity coefficient known as Matthews Correlation Coefficient, as a loss function for training the network. Moreover, we compare our loss function against other pre-existing loss functions used for semantic segmentation.

The remainder of the text is organized as follows. In Section 2, we briefly describe some previous work related to semantic segmentation of medical images using CNNs, as well as approaches related to loss function formulations for semantic segmentation. In Section 3, we present our binary semantic segmentation pipeline and our proposed loss function. In Section 4, we described some prediction metrics obtained after training the pre-existing 3D CNN model and compare them against our metric. Finally, in Section 5, we present some conclusions about our work.

2 Related Work

Regarding CNN architecture proposals for medical image semantic segmentation, the U-Net [8] is an architecture for segmenting 2D microscopy images based on a contracting path composed of a series of convolutions and pooling, and an expansive path for upsampling the feature maps obtained on the contracting path and improving the final segmentation.

Some extensions based on the U-Net architecture have been proposed for 3D medical image segmentation. For example, the 3D U-Net [9], proposed for semi and fully automated dense segmentation of the kidney, the V-Net [10], for segmenting MRI prostate images, brain lesion segmentation [11], liver and heart segmentation [12] and ultrasound segmentation [13]. It is worth mentioning that many of the aforementioned architectures have a Fully Convolutional Neural Network (FCN) architecture.

Given the problem of class imbalance in medical image datasets, apart from the different architecture proposals, there is research on the loss function formulation for specifically addressing it, exploring other options different from the widely-used cross-entropy loss function for classification. Some weighted versions of the cross-entropy have been proposed [8, 9, 14, 15] for manually weighting the contribution of pixels belonging to a certain class.

A loss function, called Malis loss, for the binary segmentation problem of separating membranes from background in 2D cell images was proposed [16]. In addition, the Tversky index is used as a loss function for image segmentation in a 3D FCN [17].

Another interesting approach is to adapt a similarity coefficient as a loss function for semantic segmentation. A similarity coefficient is used to measure the spatial overlap between a prediction and its ground truth. For example, the Dice coefficient or Dice score was adapted as a loss function for handling class imbalance [10].

Some modifications [13, 18] to the loss function proposed by Milletari et al. [10] were introduced for the binary and multi-class cases. Moreover, the same idea of adapting a similarity coefficient as a loss function for a segmentation network was used by Berman et al. [19].

3 Proposed Volumetric Image Segmentation

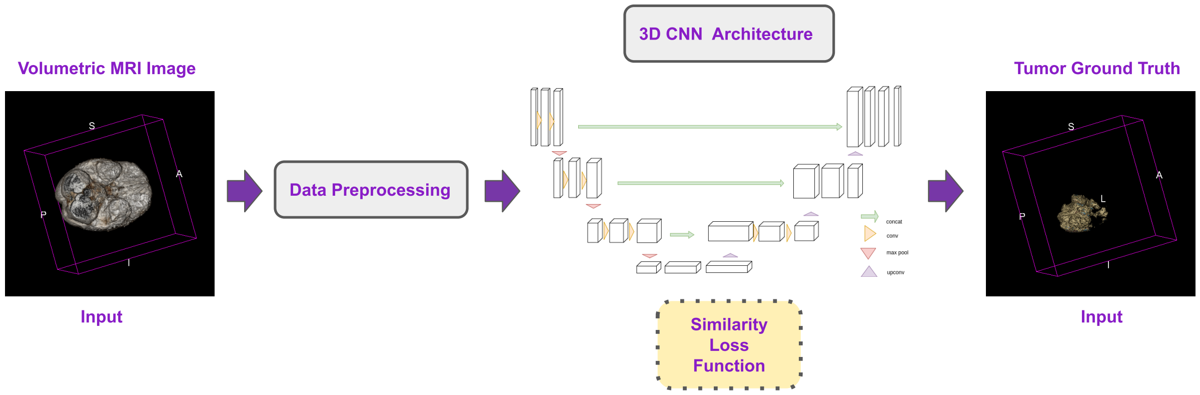


Figure 1: Volumetric image segmentation: training pipeline. Input MRI Volumetric medical image samples extracted from the BRATS15 dataset.

The pipeline that describes the building blocks of the entire network training process for our proposal is illustrated in Figure 1. Our first input is an MRI volumetric image that goes through some image pre-processing steps before feeding it to a pre-existing Three-Dimensional Convolutional Neural Network (3D CNN) architecture, trained using our loss function. The second input corresponds to the manual ground truth that our network must learn how to segment.

3.1 Data Preprocessing

Data preprocessing refers to the steps our data undergoes before feeding it to the network for training:

1. Resize: in our case, resizing means scaling all our volumetric images to a certain dimension. This is important because in volumetric image datasets, we usually find samples differing in height, width and depth (number of slices).

2. Normalization: used for mapping our data to a uniform scale. For instance, it is necessary when the image pixel values are on widely different scales.

3. Data Augmentation: for training a medical image semantic segmentation network, we need a label annotation for every single pixel in every image of the training dataset. That is why there is a lower amount of annotated data for segmentation. To tackle this problem, data augmentation techniques are used for incrementing the number of training data samples. These techniques include elastic deformation operations, rotation and translation operations [20–22]. These aim to introduce different types of noise to the original images, generating new images that will be added as part of the training set.

3.2 Matthews Correlation Coefficient (MCC)

We propose the use of the MCC metric as a loss function for guiding the binary semantic segmentation learning process of volumetric medical images. We then use it for guiding the training of a 3D CNN that performs binary classification for every pixel in our medical images.

3.2.1 Origins of MCC

MCC was introduced by biochemist Brian Matthews in [23]. In his paper, predictions of the secondary structure of T4 phage lysozyme were made in order to find a better metric to accurately measure the level of agreement between predictions and observations of helical and non helical residues (binary prediction) using the concept of correlation [23] (see Equation (1)), where G_n and P_n represent the ground truth and prediction values for sample n respectively. $\overline{G} = \frac{\sum_n G_n}{N}$ represents the fraction of observed helical samples, and $\overline{P} = \frac{\sum_n P_n}{N}$ corresponds to the fraction of predicted helical samples.

$$C = \frac{\sum_n \left(G_n - \overline{G}\right)\left(P_n - \overline{P}\right)}{\sqrt{\sum_n \left(G_n - \overline{G}\right)^2 \sum_n \left(P_n - \overline{P}\right)^2}} \tag{1}$$

The MCC formula is obtained by reducing Equation (1) for binary variables, where the ground truth and prediction are either 1 or 0. We can reduce the numerator as shown in Equation (2). Here, $\sum_n G_n P_n$ represents the intersection between ground truth and prediction, also known as the True Positive (TP):

$$\begin{aligned}
\sum_n \left(G_n - \overline{G}\right)\left(P_n - \overline{P}\right) &= \sum_n \left(G_n - \overline{G}\right)\left(P_n - \overline{P}\right) \\
&= \sum_n \left(G_n P_n - G_n \overline{P} - \overline{G} P_n + \overline{GP}\right) \\
&= \sum_n G_n P_n - \sum_n G_n \overline{P} - \sum_n \overline{G} P_n + \sum_n \overline{GP} \\
&= TP - \overline{P} \sum_n G_n - \overline{G} \sum_n P_n + n\overline{GP} \\
&= TP - N\overline{GP} - N\overline{GP} + N\overline{GP} \\
&= TP - N\overline{GP}
\end{aligned} \tag{2}$$

Both summations present in the denominator can also be reduced as shown in Equations (3) and (4). Since our variables are binary, we know that $G_n^2 = G_n$ and $P_n^2 = P_n$.

$$\begin{aligned}
\sum_n \left(G_n - \overline{G}\right)^2 &= \sum_n \left(G_n^2 - 2G_n\overline{G} + \overline{G}^2\right) \\
&= \sum_n G_n^2 - \sum_n 2G_n\overline{G} + \sum_n \overline{G}^2 \\
&= \sum_n G_n - 2\overline{G} \sum_n G_n + N\overline{G}^2 \\
&= N\overline{G} - 2N\overline{G}^2 + N\overline{G}^2 \\
&= N\overline{G} - N\overline{G}^2 \\
&= N\overline{G}(1 - \overline{G})
\end{aligned} \tag{3}$$

$$\begin{aligned}
\sum_n \left(P_n - \overline{P}\right)^2 &= \sum_n \left(P_n^2 - 2P_n\overline{P} + \overline{P}^2\right) \\
&= \sum_n P_n^2 - \sum_n 2P_n\overline{P} + \sum_n \overline{P}^2 \\
&= \sum_n P_n - 2\overline{P} \sum_n P_n + N\overline{P}^2 \\
&= N\overline{P} - 2N\overline{P}^2 + N\overline{P}^2 \\
&= N\overline{P} - N\overline{P}^2 \\
&= N\overline{P}(1 - \overline{P})
\end{aligned} \tag{4}$$

Replacing Equations (2), (3) and (4) into Equation (1) results in the final MCC formula proposed in [23], expressed as:

$$\begin{aligned}
\text{MCC} &= \frac{TP - N\overline{GP}}{N\sqrt{\overline{GP}\left(1 - \overline{G}\right)\left(1 - \overline{P}\right)}} \\
&= \frac{TP - N\overline{GP}}{\sqrt{N\overline{G}\left(1 - \overline{G}\right)N\overline{P}\left(1 - \overline{P}\right)}} \\
&= \frac{TP/N - \overline{GP}}{N\sqrt{\overline{GP}\left(1 - \overline{G}\right)\left(1 - \overline{P}\right)}}
\end{aligned} \tag{5}$$

Therefore, we can see MCC as a discretization of the Pearson Correlation Coefficient for binary variables [24]. There has been recent attention drawn to the MCC, comparing its performance to some metrics such as accuracy, precision, recall, F1 score and to the Area under ROC curve (AUC).

- Accuracy: Fraction of all correctly classified samples:

$$\frac{TP+TN}{TP+FP+FN+TN} \tag{6}$$

- Precision: Fraction of correctly classified samples among all predicted samples:

$$\frac{TP}{TP+FP} \tag{7}$$

- Recall: Fraction of correctly classified samples among correct samples:

$$\frac{TP}{TP+FN} \tag{8}$$

- F1 score: Harmonic mean between precision and recall:

$$2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

These metrics expressed in Equations (6), (7), (8) and (9) are very popular in machine learning for evaluating binary classification problems. Also, according to their definition, both precision and recall take into account just two of the four metrics present in the confusion matrix, whereas F1 score takes 3 of these metrics.

We now show some examples for analyzing how well these metrics behave under three different scenarios in a binary classification context, using the values from the confusion matrix (see Table (1)). We use a set of 10 subjects: 2 cats and 8 dogs. In the first scenario, we have the results of a classifier that labels all samples with the target class. In the second scenario, the classifier labels of all samples as belonging to the opposite class, and in the third scenario, the classifier correctly predicts the labels for all subjects. We then switch the target class for every scenario and analyze how this change affects the values produced for every metric.

In Tables (2), (3) and (4), we show the results of choosing cat as the target class. We then switch target class to dog, and the values obtained are shown in Tables (5), (6) and (7).

Table 1: Confusion matrix calculation.

	Predicted Cat	Predicted Dog	TOTAL
Observed Cat	TP	FN	$TP + FN$
Observed Dog	FP	TN	$FP + TN$
TOTAL	$TP + FP$	$FN + TN$	$TP+FN+FP+TN$

Table 2: Scenario I-A: All cats. Accuracy: 0.2, Precision: 0.2, Recall: 1, F1 score: 0.33, MCC: 0.

	Predicted Cat	Predicted Dog	Total
Observed Cat	2	0	2
Observed Dog	8	0	8
Total	10	0	10

Table 3: Scenario II-A: All dogs. Accuracy: 0.8, Precision: 0, Recall: 0, F1 score: 0, MCC: 0.

	Predicted Cat	Predicted Dog	Total
Observed Cat	0	2	2
Observed Dog	0	8	8
Total	0	10	10

From these experiments, we notice that the MCC gives a more realistic metric either when samples are correctly or incorrectly labeled by the classifiers. In addition, all metrics give an accurate result of 1

Table 4: Scenario III-A: All correct. Accuracy: 1, Precision: 1, Recall: 1, F1 score: 1, MCC: 1.

	Predicted Cat	Predicted Dog	TOTAL
Observed Cat	2	0	2
Observed Dog	0	8	8
Total	2	8	10

when all samples are correctly labeled, as in Tables (4) and (7) . The disagreement between metrics occurs when we switch target class and there is an imbalance between the number of samples of both classes. For instance, in Tables (3) and (6), values for Precision, Recall and F1 score change completely, even though classification results remain the same. This occurs because of switching the target class. In other cases, such as in Tables (2) and (5), even though prediction values are the same between the two tables, Recall and F1 score metrics produce different results.

Table 5: Scenario I-B: All cats. Accuracy: 0.2, Precision: 0, Recall: 0, F1 score: 0, MCC: 0.

	Predicted Dog	Predicted Cat	Total
Observed Dog	0	8	8
Observed Cat	0	2	2
Total	0	10	10

Table 6: Scenario II-B: All dogs. Accuracy: 0.8, Precision: 0.8, Recall: 1, F1 score: 0.88, MCC: 0.

	Predicted Dog	Predicted Cat	Total
Observed Dog	8	0	8
Observed Cat	2	0	2
Total	10	0	10

Table 7: Scenario III-B: All correct. Accuracy: 1, Precision:1, Recall: 1, F1 score: 1, MCC: 1.

	Predicted Dog	Predicted Cat	Total
Observed Dog	8	0	8
Observed Cat	0	2	2
Total	8	2	10

What makes MCC different from other metrics is that it can give more realistic results on imbalanced datasets. For instance, in [25], the authors compare MCC, accuracy and F1 score under three scenarios: positively imbalanced dataset, balanced dataset and negatively imbalanced dataset, where both F1 score and accuracy are prone to generate over-optimistic measures in presence of class imbalance. On the other hand, MCC accurately measures the binary classifier performance under all three scenarios. Differently from accuracy, which is based on the estimation of the classifier's ability on the majority class, and to the F1 score that does not consider the number of samples correctly classified as negative, MCC gives its best score when the classifier correctly predicted the majority of positive classes as well as negative classes. In [24], the authors discussed that accuracy should not be used on imbalanced datasets and performed experiments with MCC as a metric for a new classifier over various synthetic imbalanced datasets. In [26], the authors compared the MCC to the Area under ROC Curve over different ratios of imbalance, finding that, even though both of them are consistent, AUC gives more accurate results when evaluating a classifier. On the other hand, in [27], the authors questioned whether ROC metrics could be trusted on imbalanced classification scenarios as it can give a misleading overview of the performance of the classifier.

3.2.2 Loss Function Proposal: MCC as a Loss Function

We propose the MCC coefficient as a loss function since it can effectively measure the spatial overlap between the prediction and its ground truth under different scenarios of imbalance. For our medical image binary segmentation context, the fact that the MCC measures spatial overlap is convenient, when the number of pixels corresponding to the structure to be segmented is imbalanced compared to the number of background pixels.

The MCC formula given in Equation (5) can also be written in terms of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) as in Equation (10).

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (FN + TN) \times (FP + TN) \times (TP + FN)}} \tag{10}$$

In order to better understand these concepts and how they are calculated in a segmentation context, we show a graphical example for a 2D case in Figure 2. The same intuition can be extended for 3D objects.

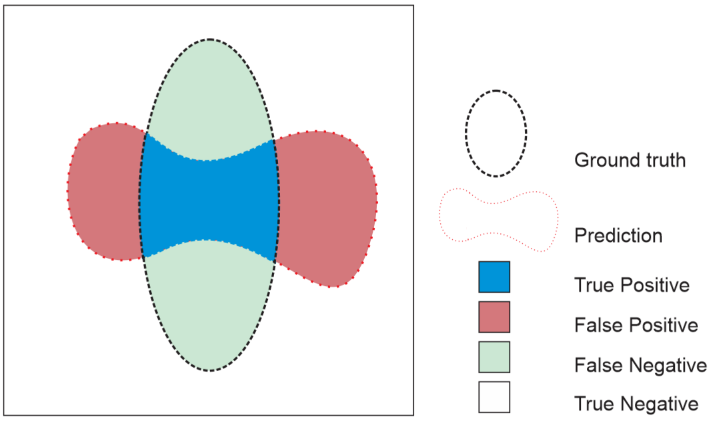


Figure 2: θ_{TP}, θ_{TN}, θ_{FP} and θ_{FN} representation in 2D.

The MCC, expressed in Equation (10), cannot be directly applied as a loss function for training a neural network, therefore, we show an adaptation for the calculation of each of its terms in Equations (11), (12), (13) and (14), in terms of operations between the prediction vectors containing the scores after softmax, and the ground truth vectors. We can then replace these values into Equation (10) for calculating the MCC.

$$TP = \sum_{n=1}^{N} \sum_{k=1}^{K} P_k^n \times G_k^n \tag{11}$$

$$TN = \sum_{n=1}^{N} \sum_{k=1}^{K} (1 - P_k^n) \times (1 - G_k^n) \tag{12}$$

$$FP = \sum_{n=1}^{N} \sum_{k=1}^{K} P_k^n \times (1 - G_k^n) \tag{13}$$

$$FN = \sum_{n=1}^{N} \sum_{k=1}^{K} (1 - P_k^n) \times G_k^n \tag{14}$$

where N is the total number of pixels, K is the total number of classes being segmented ($K = 2$ in this case), P_k^n represents the predicted probability score P of the nth pixel for belonging to class k, and similarly, G_k^n represents the ground truth score G of pixel n for class k.

The values obtained in Equation (10) lie in the range of $[-1, 1]$, where a score of 1 represents a perfect match between prediction and segmentation, and a score of -1 a total mismatch. We rescale these values in the range of $[0, 1]$ for simplicity. During training, we maximize this coefficient, whereas we minimize the loss, calculated as the complement of MCC, as in Equation (15). If our prediction is equal to its ground truth, the MCC value will reach its maximum value of 1, and after replacing this value in Equation (15), we will get a loss of 0.

$$\arg\min_{w} L = 1 - \text{MCC} \tag{15}$$

Finally, the network parameters w can be optimized using stochastic gradient descent.

4 Experiments

In this section, we describe some available loss functions, the datasets used in our experiments, as well as the segmentation results.

4.1 Pre-Existing Loss Functions

We employed the cross-entropy loss, which is very popular for classification tasks, the Dice loss [10] and the Generalized Wasserstein Dice loss [18], which were previously proposed as loss functions for semantic segmentation. The quality of the obtained segmentation results compared to their ground truth is measured in terms of the Dice score.

In this work, we adopt the 3D Fully Convolutional Neural Network architecture Dense V-Net, recently proposed by Gibson et al. [7] for training our models. When choosing a neural network architecture for our experiments, we picked one that proved to perform well on a semantic segmentation task in the medical context. For this matter, Dense V-Net was applied for the segmentation of organs in the abdominal part of the body, and compared its results beating state-of-the-art architectures, such as the baseline architecture of the V-Net [10], VoxResNet [28] and a MALF-based method [29,30].

4.2 Datasets

In this subsection, we briefly describe the datasets used in our experiments.

4.2.1 PROMISE12 Prostate Dataset

This is a dataset[1] for prostate segmentation on MRI. It consists of 50 MRI volumetric images and their corresponding ground truth, which are manual segmentations of the prostate.

These volumetric images are stored in an MHD/RAW format, where the `.mhd` files contain the headers with the image metadata, and the `.raw` files contain the image information consisting on 0 and 1 values for black and white pixels, respectively. The volumetric image size of every sample in the dataset is variable, as some images can have different dimensions from the rest. For example, we can find volumetric images with dimensions $320x320x24$, $512x512x35$, etc. In Figure 3, we show 4 slices of an MRI volumetric image (`Case29`) and their corresponding ground truth images.

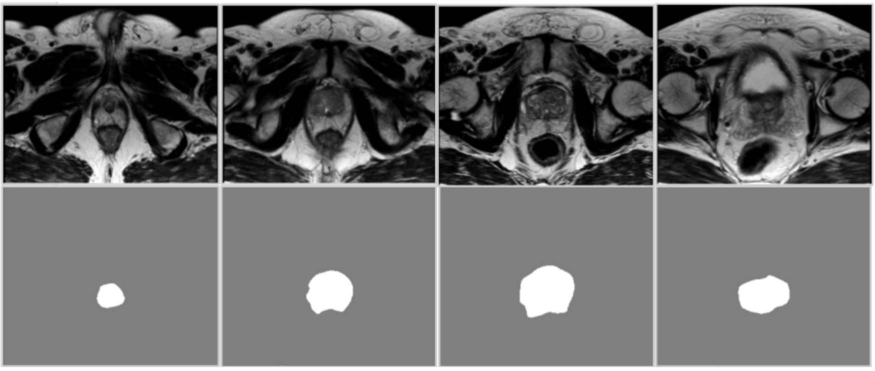


Figure 3: Example of slices from PROMISE12 MRI volumetric image (first row) and their corresponding manual ground truth (second row).

4.2.2 BRATS15 Brain Tumor Dataset

The BRATS15[2] repository is a segmentation dataset for MRI scans of brains containing Low-Grade Gliomas (LGGs) or primary brain tumors, and High-Grade Gliomas (HGGs), also known as highly malignant tumors, along with their manual segmentation images.

The dataset contains 274 samples, each one containing 4 types of volumetric images: Flair, T1, T1c and T2. They all represent the same brain image but they differ on their different levels of brightness and contrast. We used all four images for every subject during training. Dimensions for every subject are $240x240x155$ pixels.

In Figure 4, we show some slices corresponding to the MRI Flair volumetric image of `HGG1.35663` sample, in sagittal orientation (first row) and the tumor ground truth (second row). We should note that pixels from the tumor ground truth have different colors, as they represent the tumor's different stages of development, such as enhancing tumor, non-enhancing tumor, edema and necrosis.

[1] `https://promise12.grand-challenge.org`

[2] `https://www.smir.ch/BRATS/Start2015`

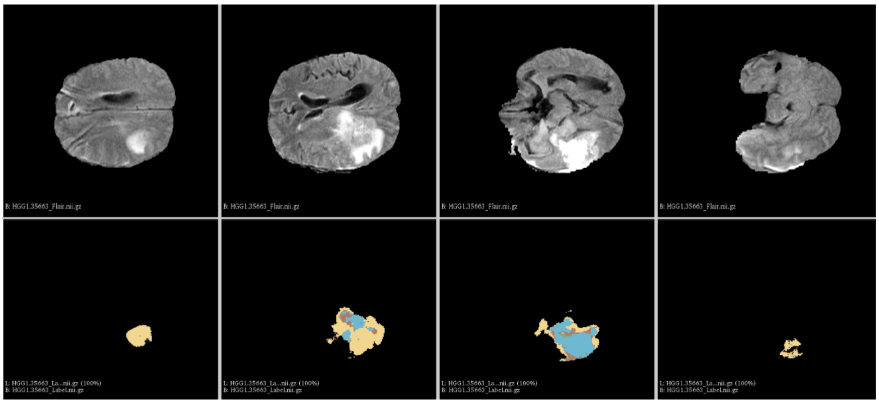
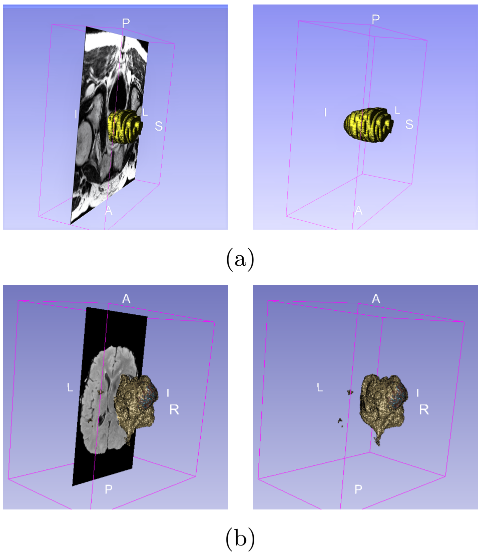

B: HGG1.35663_Flair.nii.gz
B: HGG1.35663_Flair.nii.gz
B: HGG1.35663_Flair.nii.gz
B: HGG1.35663_Flair.nii.gz
L: HGG1.35663_La...nii.gz (100%)
B: HGG1.35663_Label.nii.gz
L: HGG1.35663_La...nii.gz (100%)
B: HGG1.35663_Label.nii.gz
L: HGG1.35663_La...nii.gz (100%)
B: HGG1.35663_Label.nii.gz
L: HGG1.35663_La...nii.gz (100%)
B: HGG1.35663_Label.nii.gz

Figure 4: Example of slices from BRATS15 MRI volumetric image and their corresponding manual ground truth.

Since we are working on a binary semantic segmentation problem, we consider all different types of tumor lesions as one, and our objective becomes classifying a pixel as belonging to the tumor or not.

In Figure 5, we show a 3D visualization of the ground truths for prostate segmentation and a brain tumor segmentation from the datasets previously described. We also show an MRI slice in the middle of these volumes, extracted from the original MRI.

In the next section, we will to show some of the details for the training process of our models.



(a)

(b)

Figure 5: 3D ground truth visual examples: (a) prostate segmentation of `Case26` subject; (b) brain tumor segmentation of `HGG640963` subject.

4.3 Training Details

Our computational experiments were performed on the Manati cluster of the "Instituto de Investigaciones de la Amazonía Peruana" (Peruvian Amazon Research Institute)[3]. Each of node of the cluster has the following specifications: Intel(R) Xeon(R) CPU E5-2680 v4 with 2.40GHz, having two Tesla K80 GPUs with 11GB of GPU memory each. Even though we used a single node for training one model, we took advantage of the cluster by training all models, each one in a different node of the cluster. We left all models training for approximately a day.

Given that the volumetric images from both datasets did not share the same dimensions, PROMISE12 dataset images were resized to 64×64×64 pixels, and BRATS15 dataset images were resized to 144×144×24 pixels. In addition, data augmentation operations, such as rotation, scaling, and elastic deformations, were

[3] `http://iiap.org.pe/manati`

performed on the datasets as part of the original Dense V-Net architecture pipeline implementation. We did not perform any experiments without data augmentation.

We performed training using *batch size* $= 1$ due to the high memory overload, given the large number of parameters in the network.

Both for the PROMISE12 and the BRATS15 datasets, we used 70% of the samples for training, 20% for validation, and 10% for inference.

The 3D CNN model was trained end-to-end. This means that each of our models, for every dataset, was trained from scratch. We used validation loss in order to ensure our models did not underfit or overfit.

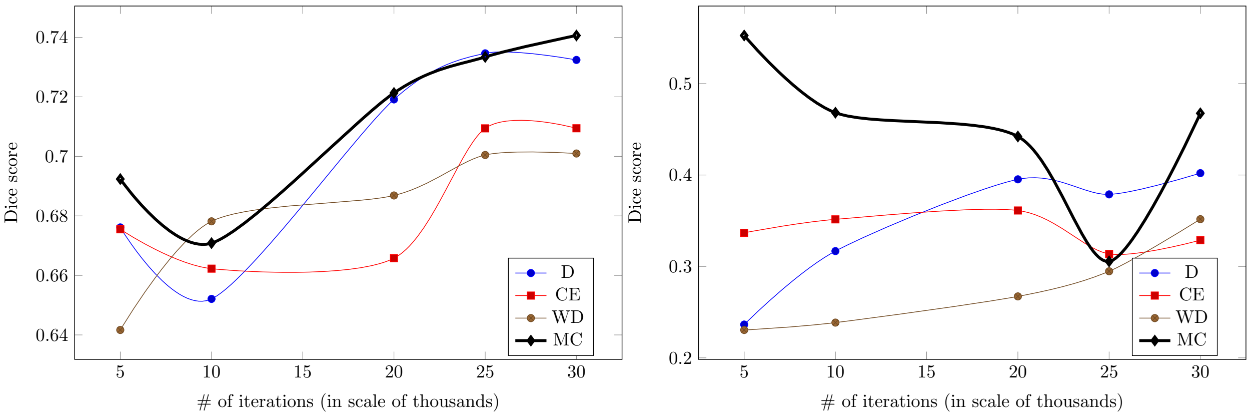
We trained four models for each dataset. Every model differs from one another on the choice of the loss function. One of the models was trained using our loss function, and we compared our results against the other three models.

4.4 Segmentation Results

For validating our loss function proposal, we conducted experiments on PROMISE12 and BRATS15 medical image segmentation datasets. We trained the same pre-existing 3D CNN architecture varying the choice of the loss function and measured the correctness of the segmentation results after performing inference with the trained models using the different loss functions. For semantic segmentation problems, the Dice score is frequently used as a metric for measuring how close the predicted segmentation volumes are to their corresponding ground truth images.

Table 8: Average Dice scores obtained with our trained models using different loss functions at training iterations 5000-30000 on the BRATS15 and PROMISE12 datasets. Our proposed loss function achieves the overall highest performance on both datasets.

BRATS15	Average Dice Scores				
Loss Function	Number of Training Iterations				
	5000	10000	20000	25000	30000
Dice	0.676	0.652	0.719	**0.735**	0.732
Cross-entropy	0.675	0.662	0.666	0.709	0.709
Wasserstein	0.642	**0.678**	0.687	0.701	0.701
Matthews	**0.692**	0.671	**0.721**	0.733	**0.741**
PROMISE12	Average Dice Scores				
Loss Function	Number of Training Iterations				
	5000	10000	20000	25000	30000
Dice	0.236	0.317	0.395	**0.379**	0.402
Cross-entropy	0.337	0.351	0.361	0.314	0.329
Wasserstein	0.23	0.239	0.267	0.295	0.352
Matthews	**0.553**	**0.468**	**0.442**	0.306	**0.468**



(a)

(b)

Figure 6: (a) Plot of the BRATS15 Dice scores shown in Table 8. (b) Plot of the PROMISE12 Dice scores shown in Table 8.

In Table 8, we show the Dice scores obtained after performing inference on the trained 3D CNN models using the different loss functions for both the BRATS15 and PROMISE12 dataset. Highest Dice scores are highlighted in boldface letters.

In Figure 6, we plot the values in Table 8 for better visualization. We use abbreviations for the Dice loss (D), Cross-entropy loss (CE), Wasserstein Dice loss (WD) and our proposed loss based on the Matthews Correlation Coefficient (MC).

For the BRATS15 dataset, our proposed loss obtained the highest Dice score during iterations 5000, 20000 and 30000. We reached the overall highest Dice score at iteration 30000. The second highest Dice score is reached with the Dice loss, and the third highest Dice score is reached again with our loss function, both at iterations 25000.

For the PROMISE12 dataset, our loss function reaches the highest Dice scores at iterations 5000, 10000, 20000 and 30000. In addition, the overall highest Dice score is reached through our loss function at iteration 5000, whereas the second highest Dice score is also reached with our loss function at iteration 30000.

4.5 PROMISE12 - 2D Visualization

We performed training on the PROMISE12 dataset during 30000 iterations. During inference, our best results were obtained with our proposed loss function after 5000 iterations, which is considerably good, given that we had to train the other models with the pre-existing loss functions during 30000 iterations to achieve similar results.

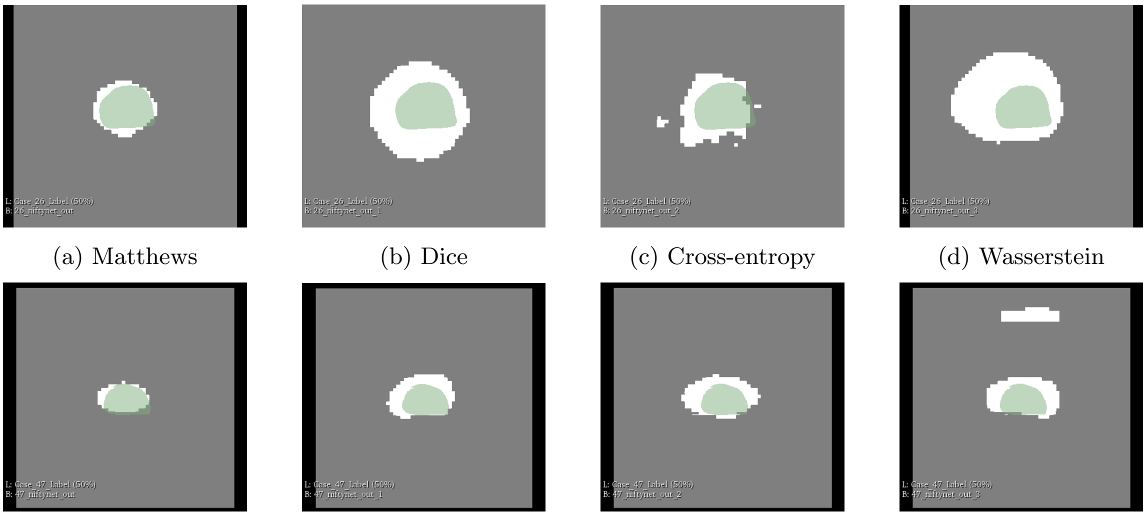
For a graphical visualization of the obtained segmentation results, we randomly selected two subjects from the PROMISE12 inference dataset: `Case26` and `Case47`.

In Figure 7, we show some slices of the obtained segmentation volumes after training the models for 5000 and 30000 iterations.

The green colored semi-transparent areas represent the prostate ground truth, whereas the white colored areas correspond to the prediction.

For segmentation results of `Case26` subject after 5000 iterations, our loss function obtains the closest segmentation volume to the ground truth volume (0.77 average Dice score), compared to the other loss functions (0.29 for the Dice loss, 0.54 for cross-entropy and 0.24 for the Wasserstein loss). The score obtained using our proposed loss function is good considering that, for this 3D semantic segmentation problem, 5000 iterations are still considered an early stage of the training process. That is why the segmentation results obtained with the other loss functions at this stage are not very accurate (many false positives), although they improve in subsequent iterations.

For segmentation results of `Case47` subject after 30000 iterations, our proposed loss function achieves the highest Dice score of approximately 0.63. The other loss functions obtain the following scores: 0.46 for the Dice loss, 0.48 for cross-entropy and 0.42 for the Wasserstein Dice loss.



(a) Matthews (b) Dice (c) Cross-entropy (d) Wasserstein

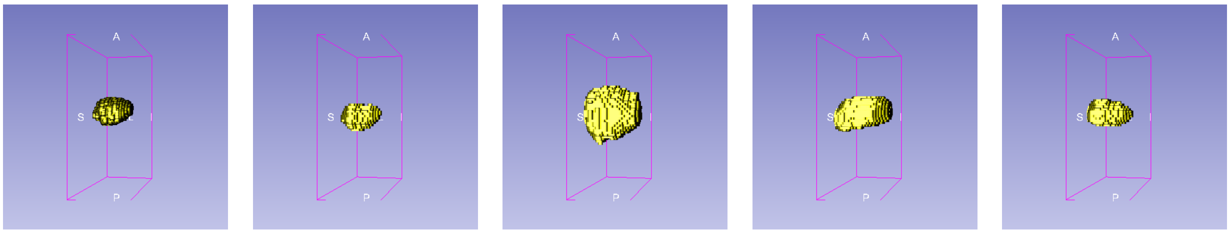
(e) Matthews (f) Dice (g) Cross-entropy (h) Wasserstein

Figure 7: Ground truth (white pixels) and predictions (semi-transparent green pixels) obtained with the trained models after iteration 5000 (a)-(d) on `Case26` subject (slice 44), and iteration 30000 (e)-(h) on `Case47` subject (slice 19).

4.6 PROMISE12 - 3D Visualization

In Figure 8, we show 3D views of the obtained segmentations by models trained after 30000 iterations for `Case26` subject, using different loss functions. For this subject, the highest Dice score is obtained with the Dice loss, followed very closely by our loss function using the Matthews coefficient. The next highest score is obtained with the Wasserstein Dice loss and its obtained prediction has a significant amount of false positives compared to the Dice loss and our loss function. Finally, cross-entropy loss function obtains the lowest score for this sample and we can observe it contains several false positives included in the resulting segmentation.

Our loss function does not achieve the highest Dice score for this sample, but it obtains the highest overall Dice score among all inference samples, as shown in Table 8.



(a) Ground truth (b) Matthews (c) Cross-entropy (d) Wasserstein (e) Dice

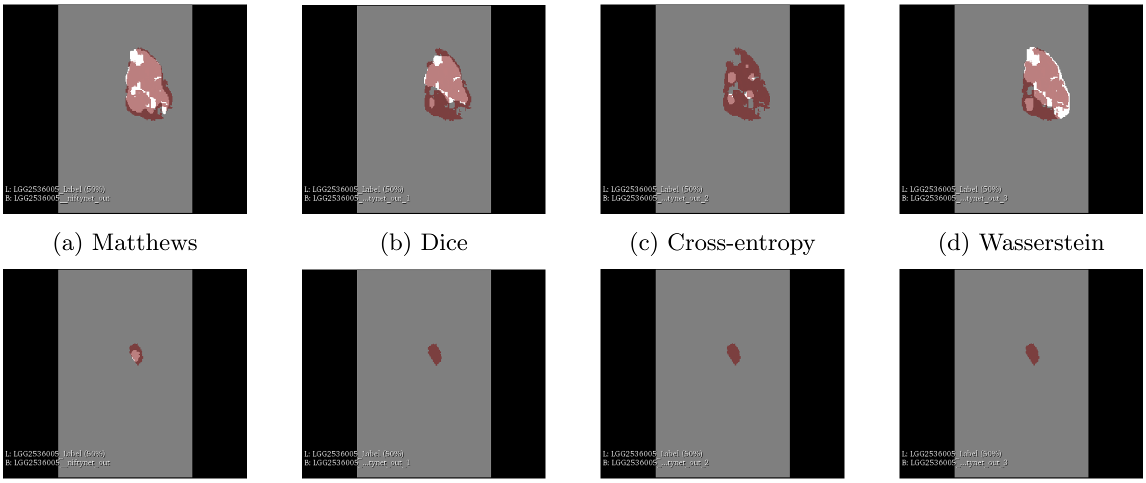
Figure 8: (a) 3D visualizations of the ground truth volume, and the segmentation volumes obtained for `Case26` subject from the PROMISE12 inference dataset, obtained with the models trained for 30000 iterations using (b) our proposed loss function, and (c), (d), (e) pre-existing loss functions.

4.7 BRATS15 - 2D Visualization

In Figure 9, we show 3 different slices extracted from the tumor prediction of `LGG2536005` subject. Ground truth is represented by the semi-transparent red pixels, whereas predictions are shown as the white pixels.

For this example, the cross-entropy loss function obtained a segmentation with an approximate 0.42 Dice score, which is the lowest among all. This is more noticeable in slices 108 and 138, since their predictions have several false negatives.

Dice and Wasserstein loss functions obtain Dice scores of 0.66 and 0.70, respectively. As we can observe in slice 97, they have a larger proportion of true positives over false positives and false negatives. However, in slice 138, we observe that the Dice loss, the cross-entropy loss and the Wasserstein loss fail to predict all of the pixels belonging to the tumor, represented by the small red area. Unlike the predictions from the pre-existing loss functions, our loss function manages to predict some of the pixels belonging to the tumor.


L: LGG2536005_label (50%)
B: LGG2536005_niftynet_out
L: LGG2536005_label (50%)
B: LGG2536005_...tynet_out_1
L: LGG2536005_label (50%)
B: LGG2536005_...tynet_out_2
L: LGG2536005_label (50%)
B: LGG2536005_...tynet_out_3

(a) Matthews (b) Dice (c) Cross-entropy (d) Wasserstein

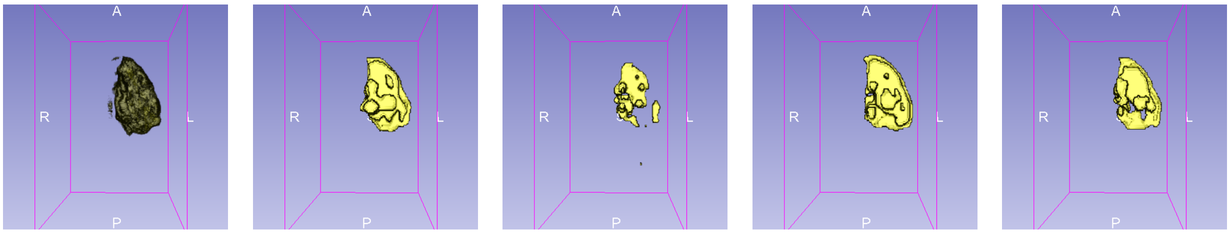
(e) Matthews (f) Dice (g) Cross-entropy (h) Wasserstein

Figure 9: Ground truth (white pixels) and predictions (semi-transparent red pixels) obtained with the trained models after iteration 30000 using the different loss functions on `LGG2536005` subject from the BRATS15 inference dataset. Slice numbers: 97 (a)-(d) and 138 (e)-(h).

Our loss function achieves a Dice score of 0.72 for this subject and, as we can see in slices 97 and 108, the segmentation is not perfect, since it has some false positive and false negative pixels, but a high rate of true positives.

4.8 BRATS15 - 3D Visualization

Finally, we show some 3D views of the segmentation results illustrated in Figure 9. 3D visualizations are shown in Figure 10. We can visually verify that the cross-entropy loss function produces the most inaccurate segmentation results. For this sample, our loss function produces the highest Dice score.



(a) Ground truth (b) Matthews (c) Cross-entropy (d) Wasserstein (e) Dice

Figure 10: 3D visualizations of the ground truth volume (a), and the segmentation volumes obtained for `LGG2536005` subject from the BRATS15 inference dataset, obtained by the models trained for 30000 iterations using (b) our proposed loss function, and (c), (d), (e) pre-existing loss functions.

5 Conclusions

We addressed the task of improving the binary semantic segmentation of medical volumetric images using a 3D CNN architecture. We proposed an improvement over this 3D CNN architecture by an adaptation of a metric, referred to as Matthews Correlation Coefficient, as a loss function for training. This loss function worked well under a class imbalance context. We compared the performance of our loss function against three other pre-existing loss functions for semantic segmentation.

Some final remarks on our work are outlined as follows:

- Deep neural networks require lots of training data in order to deliver good results. Despite of that, we showed that we could obtain good segmentation results on small segmentation datasets making use of data augmentation techniques.
- Even though our models did not produce perfect segmentations for both the PROMISE12 and BRATS15 datasets, that could be certainly improved gathering more training examples as well as choosing a different CNN architecture that produces better segmentations for each specific dataset. This is because there is not an architecture that achieves the best performance on every dataset, but our goal was to pick one that had a good performance to carry out our experiments.
- We demonstrated that we could derive a similarity-based loss function from a correlation coefficient, and that in our experiments it performed better than typical classification loss functions, such as cross-entropy for segmentation tasks. We efficiently measured the degree of overlap between prediction and ground truth, even when the classes of the pixels were imbalanced. In addition, it is important to mention that similarity-based coefficients use the context of a pixel, which is, in fact, its surrounding pixels.
- We showed that it is possible to take an existing similarity function that served for producing a metric of similarity between two objects and adapted it as an error function. We can then back-propagate the computed error through the network.
- We performed training during 30000 iterations of our models using training and validation loss as a guide to monitor our models did not overfit. During training, we devised that, even though our training and validation losses kept decreasing, sometimes our loss plots had spikes. We attribute this behavior for the batch size of 1 during training (stochastic gradient descent), since the network does not use all the training data at once. Usually, when using larger batches, loss plots are smoother. This could also be influenced by the optimization choice.
- Our proposed loss function achieved the overall highest Dice score for its obtained predictions over the two evaluated medical image segmentation datasets.

Acknowledgments

We would like to thank the financial support from Cienciactiva of the National Council for Science, Technology and Technological Innovation - CONCYTEC, Peru (grants 234-2015-FONDECYT Master's Program and 142-2015-FONDECYT). We would also like to thank FAPESP (grant #2017/12646-3) and CNPq (grant #309330/2018-7), Brazil, for their partial financial support.

References

[1] T. Moraes, P. Amorim, J. Silva, and H. Pedrini, "Isosurface Rendering of Medical Images Improved by Automatic Texture Mapping," in *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, 2018, pp. 379–385. [Online]. Available: https://doi.org/10.1080/21681163.2016.1254069

[2] ——, "Web-Based Interactive Visualization of Medical Images in a Distributed System," in *14th International Conference on Computer Graphics Theory and Applications*, Prague, Czech Republic, Feb. 2019, pp. 346–353. [Online]. Available: 10.5220/0007626103460353

[3] P. Amorim, T. Moraes, J. Silva, and H. Pedrini, "An Out-of-Core Volume Rendering Architecture," in *IV ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, Oct. 2013, pp. 173–179.

[4] T. Moraes, P. Amorim, J. Silva, H. Pedrini, and M. Meurer, "Medical Volume Rendering based on Gradient Information," in *V ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, Tenerife, Canary Islands, Spain, Oct. 2015, pp. 181–186. [Online]. Available: 10.1201/b19241-31

[5] J. Bobadilla and H. Pedrini, "Lung Nodule Classification based on Deep Convolutional Neural Networks," in *21st Iberoamerican Congress on Pattern Recognition*, vol. 10125, Lima, Peru, Nov. 2016, pp. 117–124. [Online]. Available: https://doi.org/10.1007/978-3-319-52277-7_15

[6] P. Amorim, T. Moraes, R. Rezende, J. Silva, and H. Pedrini, "Medical Imaging for Three-Dimensional Computer-Aided Models," in *Tissue Engineering and Regeneration: 3D Printing and Biofabrication.* Springer International Publishing, 2018, pp. 195–221.

[7] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Automatic Multi-Organ Segmentation on Abdominal CT with Dense V-Networks," *IEEE Transactions on Medical Imaging*, vol. 37, no. 8, pp. 1822–1834, 2018. [Online]. Available: 10.1109/TMI.2018.2806309

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2015, pp. 234–241. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28

[9] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2016, pp. 424–432. [Online]. Available: https://doi.org/10.1007/978-3-319-46723-8_49

[10] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *Fourth International Conference on 3D Vision.* IEEE, 2016, pp. 565–571. [Online]. Available: 10.1109/3DV.2016.79

[11] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation," *Medical Image Analysis*, vol. 36, pp. 61–78, 2017. [Online]. Available: https://doi.org/10.1016/j.media.2016.10.004

[12] Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, and P.-A. Heng, "3D Deeply Supervised Network for Automated Segmentation of Volumetric Medical Images," *Medical Image Analysis*, 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.05.001

[13] X. Yang, L. Yu, S. Li, X. Wang, N. Wang, J. Qin, D. Ni, and P.-A. Heng, "Towards Automatic Semantic Segmentation in Volumetric Ultrasound," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2017, pp. 711–719. [Online]. Available: https://doi.org/10.1007/978-3-319-66182-7_81

[14] P. F. Christ, F. Ettlinger, F. Grün, M. E. A. Elshaera, J. Lipkova, S. Schlecht, F. Ahmaddy, S. Tatavarty, M. Bickel, and P. Bilic, "Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks," *arXiv Preprint arXiv:1702.05970*, 2017.

[15] A. Casamitjana, S. Puch, A. Aduriz, E. Sayrol, and V. Vilaplana, "3D Convolutional Networks for Brain Tumor Segmentation," *MICCAI Challenge on Multimodal Brain Tumor Image Segmentation (BRATS)*, pp. 65–68, 2016.

[16] F. Tschopp, J. N. Martel, S. C. Turaga, M. Cook, and J. Funke, "Efficient Convolutional Neural Networks for Pixelwise Classification on Heterogeneous Hardware Systems," in *IEEE 13th International Symposium on Biomedical Imaging.* IEEE, 2016, pp. 1225–1228. [Online]. Available: 10.1109/ISBI.2016.7493487

[17] S. R. Hashemi, S. S. M. Salehi, D. Erdogmus, S. P. Prabhu, S. K. Warfield, and A. Gholipour, "Tversky as a Loss Function for Highly Unbalanced Image Segmentation using 3D Fully Convolutional Deep Networks," *CoRR*, vol. abs/1803.11078, 2018. [Online]. Available: http://arxiv.org/abs/1803.11078

[18] L. Fidon, W. Li, L. C. Garcia-Peraza-Herrera, J. Ekanayake, N. Kitchen, S. Ourselin, and T. Vercauteren, "Generalised Wasserstein Dice Score for Imbalanced Multi-class Segmentation using Holistic Convolutional Networks," *arXiv preprint arXiv:1707.00478*, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-75238-9_6

[19] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The Lovász-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-over-Union Measure in Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421. [Online]. Available: 10.1109/CVPR.2018.00464

[20] Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V. C. Mok, L. Shi, and P.-A. Heng, "Automatic Detection of Cerebral Microbleeds from MR Images via 3D Convolutional Neural Networks," *IEEE Transactions Medical Imaging*, vol. 35, no. 5, pp. 1182–1195, 2016. [Online]. Available: 10.1109/TMI.2016.2528129

[21] T. Brosch, L. Y. Tang, Y. Yoo, D. K. Li, A. Traboulsee, and R. Tam, "Deep 3D Convolutional Encoder Networks With Shortcuts for Multiscale Feature Integration Applied to Multiple Sclerosis Lesion Segmentation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1229–1239, 2016. [Online]. Available: 10.1109/TMI.2016.2528821

[22] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240–1251, 2016. [Online]. Available: https://doi.org/10.1007/s10916-019-1416-0

[23] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975. [Online]. Available: https://doi.org/10.1016/0005-2795(75)90109-9

[24] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PloS one*, vol. 12, no. 6, 2017. [Online]. Available: 10.1371/journal.pone.0177678

[25] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, p. 6, 2020. [Online]. Available: 10.1186/s12864-019-6413-7

[26] C. Halimu, A. Kasem, and S. S. Newaz, "Empirical comparison of area under roc curve (auc) and mathew correlation coefficient (mcc) for evaluating machine learning algorithms on imbalanced datasets for binary classification," in *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3310986.3311023

[27] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0118432

[28] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng, "VoxResNet: Deep Voxelwise Residual Networks for Brain Segmentation from 3D MR Images," *NeuroImage*, vol. 170, pp. 446–455, 2018. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2017.04.041

[29] M. P. Heinrich, M. Jenkinson, M. Brady, and J. A. Schnabel, "MRF-based Deformable Registration and Ventilation Estimation of Lung CT," *IEEE Transactions on Medical Imaging*, vol. 32, no. 7, pp. 1239–1248, 2013. [Online]. Available: 10.1109/TMI.2013.2246577

[30] H. Wang, J. W. Suh, S. R. Das, J. B. Pluta, C. Craige, and P. A. Yushkevich, "Multi-Atlas Segmentation with Joint Label Fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 611–623, 2012. [Online]. Available: 10.1109/TPAMI.2012.143