

Article

# Dissimilarity Corrective Generative Adversarial Network for Brain Image Segmentation

Rukesh Prajapati  and Goo-Rak Kwon \* 

Department of Information and Communication Engineering, Chosun University, 309 Pilmun-Daero, Dong-gu, Gwangju 61452, Republic of Korea

\* Correspondence: grkwon@chosun.ac.kr

**Abstract:** More accurate diagnosis of brain disorders can be achieved by properly analyzing structural changes in the brain. For the quantification of change in brain structure, the segmentation task is crucial. Recently, generative adversarial networks (GAN) have been rapidly developed and used in many fields. Segmentation of medical images with these networks will greatly improve performance. However, segmentation accuracy improvement is a challenging task. In this paper, we propose a novel corrective algorithm for updating the accuracy and a novel loss function based on dissimilarity. First, we update the generator using the typical dice similarity coefficient (DSC) as a loss function only. For the next update, we use the same image as input and obtain the output; this time, we calculate dissimilarity and update the generator again. In this way, false prediction, due to the first weight update, can be updated again to minimize the dissimilarity. Our proposed algorithm can correct the weights to minimize the error. The DSC scores obtained with the proposed algorithm and the loss function are higher, and clearly outperformed the model with only DSC as the loss function for the generator.

**Keywords:** dice similarity coefficient; dissimilarity coefficient; generative adversarial network; segmentation



**Citation:** Prajapati, R.; Kwon, G.-R. Dissimilarity Corrective Generative Adversarial Network for Brain Image Segmentation. *Appl. Sci.* **2022**, *12*, 12944. <https://doi.org/10.3390/app122412944>

Academic Editors: Alexander N. Pisarchik, Alexander E. Hramov and Victor B. Kazantsev

Received: 19 October 2022

Accepted: 12 December 2022

Published: 16 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



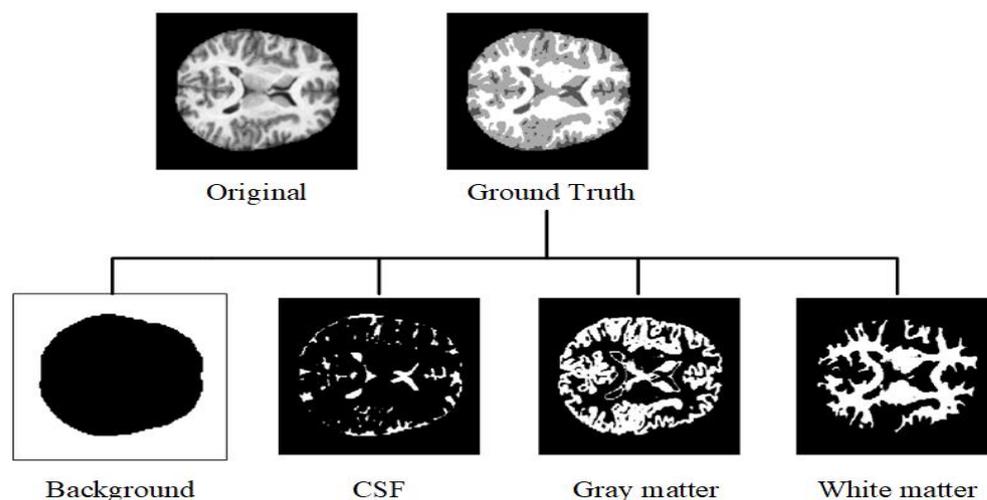
**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

By assigning a specific value to each pixel, semantic segmentation localizes the class on an image [1]. Magnetic resonance imaging (MRI) is the most widely used imaging technology to study the brain. The MRI repeats different excitations and produces a contrast image [2]. MRI can be used to diagnose Alzheimer's disease (AD) and multiple sclerosis [3]. One of the common biomarkers for the diagnosis of AD is tissue atrophy [4]. Accurate identification and categorization of brain tissue is crucial for such a diagnosis. To avoid difficult tasks, such as manually analyzing MRI datasets, automatic segmentation techniques are developed [5]. Recently, many deep learning models have been developed for the segmentation of brain tissues [6–12]. The brain comprises three types of tissues: cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM), as shown in Figure 1. As the brain MRI is a 3D image, it will consume lots of computational resources for the process. We all know that 3D images are a stack of 2D images. When 2D images are stacked, a third dimension is added which will make it 3D. For faster processing with less resources a 2D approach can be used where each slice is processed one by one until all of the slices in the 3D are processed. The result of processing 3D at once or processing and adding each slice to create whole image is the same.

Before deep learning, algorithms were implemented for the segmentation purpose. However, it takes a lot of time to obtain results from such processes [8]. Unlike deep learning, the user must change parameters and steps in the algorithm manually as it cannot learn from the data itself. The segmentation of the medical images is developed along with the progress in the deep learning models. Recently, there has been significant development

in GAN [13–17]. GAN is used for a variety of applications, such as generating videos from images [18], increasing resolution [19], and generating an image from text [20]. GAN structures comprise of two parts: a generator and a discriminator [21]. There has been a development of different types of discriminators. Unlike a typical discriminator, which only classifies the output of the generator [22], a new discriminator that discriminates the image on the level of the patch has been introduced [23]. Recent studies have shown that semantic segmentation can be improved with the GAN [22]. Unlike traditional GAN, a multi-discriminator GAN was introduced to distribute the burden for the worker machine [24].



**Figure 1.** Binary maps generated from the ground truth for different classes.

DSC is used as the evaluation metrics for most segmentation tasks. It was first proposed by Dice [25]. In the case of segmentation, DSC is basically calculated by obtaining the intersection area between the output and the ground truth. The value of DSC is always between zero and one. The higher the value of DSC, the better the result will be. To obtain the DSC of the multiclass segmentation, DSC for each class is obtained, and the average is calculated [26]. We can use this evaluation matrix as a loss function during the training too. As deep learning tries to minimize the error while training, we can use the negative of the DSC as the loss function. However, the limitation of the DSC is that it only gives information about the correctly predicted pixels.

Motivated by the multiple-discriminator GAN, we propose a novel method to update a GAN alternately with two discriminators, using DSC and a new proposed loss function. We studied the structure of the predicted output and formulated a new function that could give information about areas that are not included as the intersection in the DSC. We hypothesize that instead of using an average of all loss functions at once, a new corrective alternative update of GAN would be able to train the generator properly. In our algorithm, the generator weights will be updated twice in an iteration. The first one will be based on DSC. After that update, the same images are passed through the generator, and the output from this will be used to calculate dissimilarity. This update will be mainly focused on minimizing the dissimilarity, or the false prediction.

In this paper, we describe a new coefficient that measures the dissimilarity among the binary images and a novel algorithm to update the GAN alternately in a corrective way with two discriminators. Our main contributions are:

- We propose a new corrective way of updating weights of the generator using two discriminators alternately. The discriminator used in the first update discriminates the image at a patch level, whereas the second one discriminates the whole image at once. These different updates will be based on different loss functions;
- We propose new loss functions based on the dissimilarity of the images. We formulate the functions and define an algorithm to calculate them from an image;

- We confirm with a DSC score of the model on test data that the proposed algorithm and loss functions help the generator outperform the typical generator trained with DSC alone.

## 2. Materials and Methods

### 2.1. Data

We use the Open Access Series of Imaging Studios 1 (OASIS 1) dataset [27]. For training, we select 60 subjects randomly. The MRI scan of each subject has dimensions of  $176 \times 208 \times 176$ . There are three planes (axial, coronal, and sagittal). We extract, train, and test 2D images for each plane individually. The extracted images are resized to the dimension of  $256 \times 256$  before training and testing. In the MRI scan of the brain, there are empty spaces outside of the brain. While extracting 2D images from those MRI, images at the edge are blank and have no information about the brain. We remove those blank images and consider only images with the brain portion or information. Visually analyzing the MRI scans, we determine the range of number of slices used in each plane. For the axial plane, we use slices from 15 to 145. Similarly, we use slices from 30 to 180 in the coronal plane, and 25 to 145 in the sagittal plane. All together, we used 7800, 9000, and 7200 images for axial, coronal, and sagittal planes, respectively, while training the model individually. The ground truth image provided in the OASIS 1 dataset are produced using Markov random field model and an associated Expectation-Maximization algorithm [27].

### 2.2. Method

The following equation provides the objective of a conditional GAN:

$$\mathcal{L}_cGAN(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \tag{1}$$

where a generative model  $G$  tries to minimize  $\mathcal{L}_cGAN(G, D)$ , and the discriminator  $D$  tries to maximize it [13]. Mixing GAN objectives and a more traditional loss have been found beneficial [28]. The task of the generator will be to fool the discriminator and produce segmentation as near as possible to the ground truth. In contrast, the job of the discriminator is unchanged [27].

We aim to train our GAN network with two discriminators, where the generator is updated twice in an iteration according to the DSC as a loss function in first update, followed by the update with respect to the new proposed dissimilarity coefficients as a loss function. The updates of the generator according to the discriminators are illustrated in Figure 2. We use different discriminators for the discrimination of the generator’s output at a different update. The process of calculating dissimilarity coefficients is summarized in Algorithm 1 and the process of updating the generator is summarized in Algorithm 2.

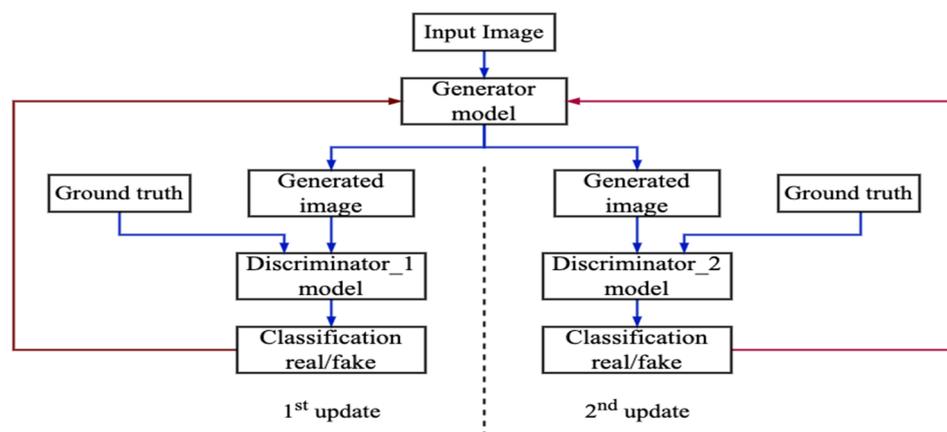


Figure 2. Illustration of updating the generator twice in an iteration with a different discriminator at each time.

During the first update process, the generator model produces a segmented output for the corresponding input. DSC is used as a loss function. As the value of DSC is always between 0 and 1, the bigger the value, the better will be the result. So, we use the negative of the DSC calculated as the loss function, and then a patchGAN discriminator penalizes at the patch level. The loss  $l^{u1}$  for the first update is the weighted sum of a DSC loss and an adversarial loss:

$$l^{u1} = l_{DSC} + 10^{-3}l_{Gen} \quad (2)$$

where  $l_{DSC}$  is negative of DSC value, which can be calculated as:

$$DSC = \frac{2|I \cap \hat{I}|}{|I| + |\hat{I}|} \quad (3)$$

where  $I$  is the ground truth, and  $\hat{I}$  is the predicted output. The second update is to train the model about giving information and minimizing the false prediction during the first update. After updating the generator, we predict the same image through the generator and calculate the dissimilarity between the ground truth and the predicted image. We propose two types of coefficients related to dissimilarity. The dissimilarity coefficient (DC) is related to the false positive and false negative predictions. The calculation of these coefficients is given in Equations (7) and (9). The total DC loss is the sum of these two coefficients. The second update is focused on training the generator to correct and minimize the dissimilarity. The loss  $l^{u2}$  for the second update is the weighted sum of a DC loss and an adversarial loss:

$$l^{u2} = l_{DC} + 10^{-3}l_{Gen} \quad (4)$$

For the second update, the generative loss  $l_{Gen}$  is defined based on the probabilities of the discriminator over all training data as:

$$l_{Gen} = \sum_{n=1}^N -\log D_{\theta_D}(G_{\theta_G}(I)) \quad (5)$$

where,  $G_{\theta_G}$  and  $D_{\theta_D}$  are generator and discriminator networks parametrized by  $\theta_G$  and  $\theta_D$ , respectively.

### 2.2.1. UNet as Generator

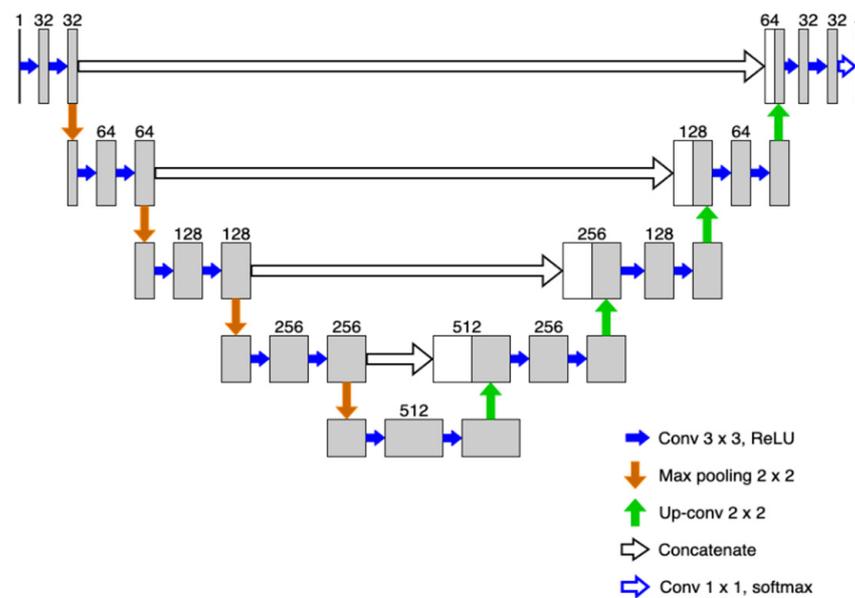
For medical image segmentation, UNet architecture is the most widely used [29]. We also used UNet as the generator model for our proposed model, which is illustrated in Figure 3. UNet architecture has an encoder and decoder path. The encoder comprises of convolution layers and max pooling, whereas the decoder comprises of convolution layers and up convolution. The feature map from the encoder is concatenated with the corresponding feature map of the decoder, using a skip connection. We use two convolution layers followed by batch-normalization layers and LeakyReLU as the activation function in each step of the encoder. Each step is followed by  $2 \times 2$  max pooling. In the decoder path, each up sampling is followed by two convolution layers as in the encoder. The output of up sampling is concatenated with the feature map from the encoder before it goes through the convolution.

### 2.2.2. Multiple Discriminators

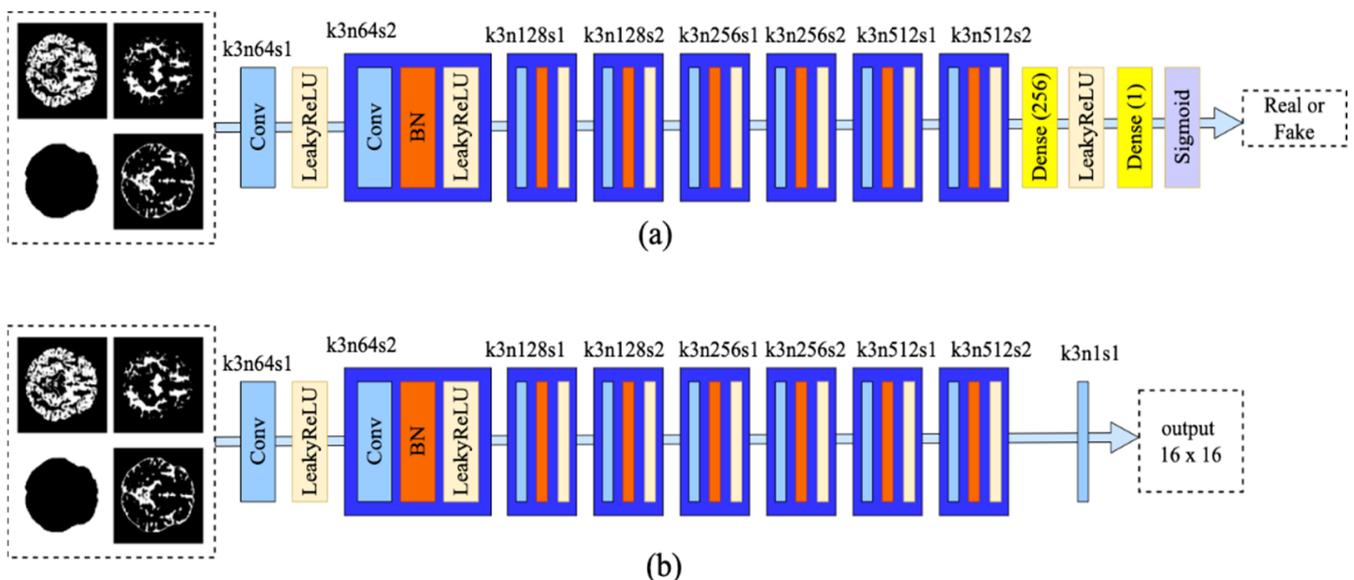
We performed a study based on two discriminators. One discriminator is a conditional GAN which discriminates real segmented images from generated segmented images. Another discriminator is Markovian discriminator (patchGAN) [27].

For the discriminator based on a conditional GAN, we follow the summarized architecture by [15]. This architecture avoids the use of max pooling. The maximization problem in Equation (1) is solved by training the discriminator network [20]. There are eight convolutional layers increasing from 64 to 512 by a factor of 2; the same as in the VGG

architecture [30]. To reduce the resolution when the feature numbers are doubled, stride convolutions are used. We use LeakyReLU as the activation function ( $\alpha = 0.2$ ). The final feature maps are followed by two dense layers and then by a sigmoid activation function, which provides a probability for the classification. Figure 4a illustrates the architecture of this conditional GAN. The task of the discriminator is to examine the output of the generator and determine whether it is similar to the original ground truth or not. If it is comparably similar to the ground truth, it will predict it as “real”, otherwise it will be “fake”. Hence, the generator will consider this loss to generate images similar to the ground truth as much as possible. The discriminator in Figure 4a functions like a binary classifier where real is represented by true and fake is represented by false.



**Figure 3.** Architecture of a Generator Network with the number of channels indicated at the top of each convolution.



**Figure 4.** Architecture of the Discriminator Network of a (a) conditional GAN and (b) patchGAN with corresponding kernel size ‘k’, stride ‘s’, and a number of feature maps ‘n’ indicated for each convolutional layer.

We follow patchGAN architectural guidelines summarized by [27]. The discriminator in patchGAN only discriminates the structure at the scale of patches. A patch of size  $N \times N$  is classified as real or fake. Assuming the pixels separated by more than a patch are independent, this discriminator models the image as a Markov random field [27]. Figure 4b illustrates the architecture of this patchGAN. The structure is similar to the conditional GAN we used. This discriminator also comprises of eight convolutional layers increasing from 64 to 512 by a factor of 2, and LeakyReLU ( $\alpha = 0.2$ ) is used as an activation function. In this architecture, the final two dense layers are replaced by a single convolution layer with a single feature map with a kernel size  $3 \times 3$ . The final result of this discriminator is a of size of  $16 \times 16$ .

### 2.3. Proposed Loss Functions

In DSC, only the intersection of the predicted and the ground truth is calculated. Thus, using DSC as a loss function will only give information about the true positive pixels in the result image. There are false negative and positive pixels in the predicted output, along with the true positive pixels. This pixel information can also help to improve the intersection or the true positive pixels.

#### 2.3.1. Dissimilarity from False Negative

Studying the false negative pixels and reducing the number of those will automatically help us increase true positive pixels. The relation between false-negative pixels and DSC is inversely proportional. The lower the number of false-negative pixels, greater the value of DSC.

$$DSC \propto \frac{1}{P_{FN}} \quad (6)$$

where  $P_{FN}$  is the number of pixels of the false negative in the predicted image. Figure 5 shows the process of getting false negative pixels from the predicted output. To get false negative pixels, we first invert the predicted output image. To do so, we subtract each pixel of the output image from a integer value one. Then the inverted image is multiplied by the ground truth image. As both images (ground truth and predicted output) are binary images, only the intersection between them will not be zero. This remaining non-zero area is the false negative pixel, as this area is non-zero in the ground truth. From this pixel value, we can calculate a dissimilarity coefficient. The dissimilarity coefficient for false-negative pixels prediction is calculated as,

$$DC_{FN} = \frac{|I \cap inv(\hat{I})|}{|inv(\hat{I})|} \quad (7)$$

where  $|inv(\hat{I})|$  is the number of pixels of the ROI from the inverted predicted output.  $DC_{FN}$  is a metric of area overlap between the ground truth and the inverted predicted image.  $|I \cap inv(\hat{I})|$  represents the number of pixels that are predicted as background but are not background in the ground truth image. Thus, we consider these pixels as a false negative. The process of calculating a dissimilarity coefficient for false negative values is summarized in Algorithm 1. From Equation (7) we can say that the  $DC_{FN}$  is the ratio of the number of false negative pixels, or false background pixels, to the number of total background pixels predicted. This ratio tells us how many of the predicted background pixels do not match with the actual ground truth background pixels.

#### 2.3.2. Dissimilarity from False Positive

Among the predicted pixels, there are false positive pixels. While increasing the intersection with the ground truth, the model should also know about the false positive pixels to minimize the prediction of those pixels in the future. The relation of the number of the true positive pixels with the number of the false positive pixels is also inversely proportional. Reduction in the false positive pixels will help to focus or give information

to increase the intersection of the ground truth and the predicted region, resulting in the improvement of the DSC.

$$DSC \propto \frac{1}{P_{FP}} \quad (8)$$

where  $P_{FP}$  is the number of pixels of the false positive in the predicted image. Figure 6 shows the process of getting false positive pixels from the predicted output. To get false negative pixels, we first invert the ground truth image. To do so, we subtract each pixel of the ground truth image from a integer value one. Then the inverted image is multiplied by the predicted image. As both images (ground truth and inverted predicted output) are binary images, only the intersection between them will not be zero. This remaining non-zero area is the false positive pixel as this area is non-zero in the output image. From this pixel value, we can calculate a dissimilarity coefficient for false positive. The dissimilarity coefficient for false-positive pixels prediction is calculated as,

$$DC_{FP} = \frac{|inv(I) \cap \hat{I}|}{|\hat{I}|} \quad (9)$$

where  $|inv(I)|$  is the number of pixels of the ROI from the inverted ground truth, and  $|\hat{I}|$  is the number of pixels of the ROI from the predicted output image.  $DC_{FP}$  is a metric of area overlap between the inverted ground truth and the predicted image.  $|inv(I) \cap \hat{I}|$  represents the number of pixels that are predicted as foreground but are not foreground in the ground truth image. Thus, we consider these pixels as a false positive. The process of calculating dissimilarity coefficient for false positive values is summarized in Algorithm 1. From Equation (9) we can say that the  $DC_{FP}$  is the ratio of number of false positive pixels, or false foreground pixels, to the number of total foreground pixels, or ROI predicted. This ratio tells us how many of the predicted ROI pixels do not match with the actual ground truth ROI pixels.

---

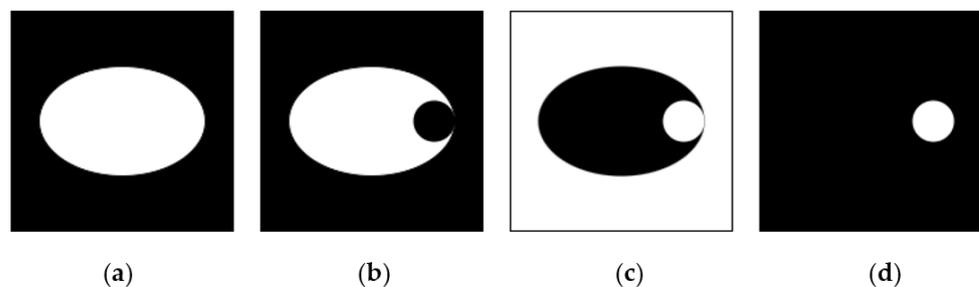
**Algorithm 1** Computation of dissimilarity coefficient for false negative  $DC_{FN}$  and false positive  $DC_{FP}$

---

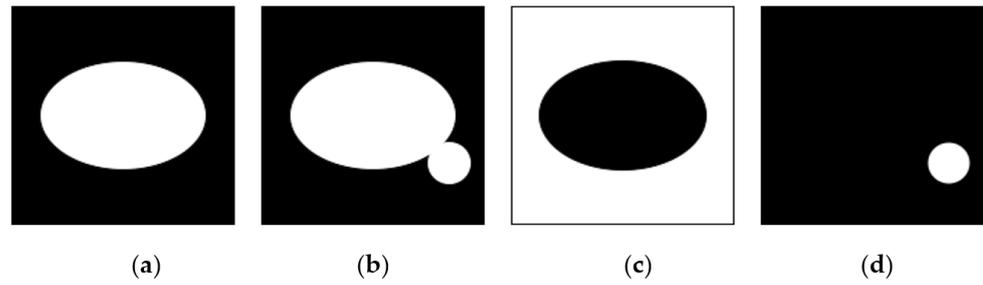
Input: ground truth image  $I$ , predicted image  $\hat{I}$

Output:  $DC_{FN}$ ,  $DC_{FP}$

1. Inverse\_ $\hat{I} \leftarrow 1 - \hat{I}$
  2. Intersection\_FN  $\leftarrow I \times$  inverse\_ $\hat{I}$
  3. union\_FN  $\leftarrow$  sum (inverse\_ $\hat{I}$ )
  4.  $DC_{FN} \leftarrow 2 * \text{Intersection\_FN} / \text{union\_FN}$
  5. Inverse\_I  $\leftarrow 1 - I$
  6. Intersection\_FP  $\leftarrow$  inverse\_I  $*$   $\hat{I}$
  7. union\_FP  $\leftarrow$  sum ( $\hat{I}$ )
  8.  $DC_{FP} \leftarrow 2 \times \text{Intersection\_FP} / \text{union\_FP}$
  9. return  $DC_{FP}$ ,  $DC_{FN}$
- 



**Figure 5.** Binary maps of: (a) the ground truth, (b) the predicted output, (c) the inverted predicted output, and (d) the false negative area obtained by multiplying (a,c).



**Figure 6.** Binary maps of: (a) the ground truth, (b) the predicted output, (c) the inverted ground truth, and (d) the false positive area obtained by multiplying (b,c).

---

**Algorithm 2** Our proposed algorithm to update generator  $G$  with two different discriminators ( $D_1$  and  $D_2$ )

---

1. for epoch do
  2. Obtain segmented image  $\hat{I}$  from input  $I$ :  $\hat{I} = G(I)$
  3. Export the ground truth  $I^{GT}$  and segmented image  $\hat{I}$  as input to discriminator  $D_1$
  4. Train  $D_1$  (update weights):  

$$\nabla_{\theta_{d1}} = \frac{1}{m} \sum_{i=1}^m [\log D_1(I^{GT}) + \log(1 - D_1(\hat{I}))]$$
  5. Train  $G$  (update weights):  

$$\nabla_{\theta_g} = \nabla_{\theta_{DSC}} + 10^{-3} * \nabla_{\theta_{gen}}$$

$$\nabla_{\theta_{gen}} = \frac{1}{m} \sum_{i=1}^m -\log D_1(G(I))$$

$$\nabla_{\theta_{DSC}} = \frac{1}{m} \sum_{i=1}^m [-2 * (|I \cap \hat{I}|) / (|I| + |\hat{I}|)]$$
  6. Obtain segmented image  $\hat{I}$  from input  $I$ :  $\hat{I} = G(I)$
  7. Export the ground truth  $I^{GT}$  and segmented image  $\hat{I}$  as input to discriminator  $D_2$
  8. Train  $D_2$  (update weights):  

$$\nabla_{\theta_{d2}} = \frac{1}{m} \sum_{i=1}^m [\log D_2(I^{GT}) + \log(1 - D_2(\hat{I}))]$$
  9. Train  $G$  (update weights):  

$$\nabla_{\theta_g} = \nabla_{\theta_{DC}} + 10^{-3} * \nabla_{\theta_{gen}}$$

$$\nabla_{\theta_{gen}} = \frac{1}{m} \sum_{i=1}^m -\log D_2(G(I))$$

$$\nabla_{\theta_{DC}} = \frac{1}{m} \sum_{i=1}^m [DC_{FP} + DC_{FN}]$$
  10. end for
- 

#### 2.4. Training Details and Parameters

We trained the model for all planes on a NVIDIA GeForce RTX 3090 GPU using images extracted from different planes individually. The MRI subjects for training are different from the subjects for the testing. We obtained inputs and the ground truth images from MRIs and rescaled them to the dimension of  $256 \times 256$  before feeding them to the network. Since we are using DSC as a loss function, we need a binary map corresponding to each class. We extracted a binary map for each class (background, CSF, GM, and WM). For that, we generated binary images which have a value of one only at the pixels where class or tissue is present in the ground truth, otherwise the value was set to 0. We used Adam [31] with  $\beta_1 = 0.9$  for the optimization. The proposed model networks were trained with a learning rate of  $10^{-3}$  as in [32]. The generator is updated twice in an iteration with different discriminators, and the discriminators are updated alternately with each update.

#### 2.5. Evaluation Metrics

We used two types of metrics for the evaluation: DSC and Volumetric Overlap Error (VOE) [33]. Using Equation (3) we can calculate the DSC scores. For the VOE, we calculate the intersection and the union of the segmented and the ground truth.

For the ground truth segmentation map  $S$  and the predicted segmentation map  $S'$ , the VOE is defined in Equation (10).

$$\text{VOE} = 1 - \frac{|S \cap S'|}{|S \cup S'|} \quad (10)$$

### 3. Results

Our main goal is to improve DSC scores by training GAN with dissimilarity loss functions and DSC as loss function. We proposed a new algorithm to update the generator with multiple discriminators instead of a single discriminator to make corrections in the updated weight. To check whether our assumption and proposed method's performance, we performed two experiments described in the section below.

#### 3.1. Single Discriminator GAN Model with DSC and DC Loss Functions (Exp\_model\_1)

In this experiment, we used a UNet generator as in the proposed algorithm. But in this case, we used a single discriminator only. During the training process, DSC and DC are calculated from the same segmented output, and the generator and discriminator are updated. The loss function for this model's generator update is:

$$l = l_T + 10^{-3}l_{Gen} \quad (11)$$

$$l_T = l_{DSC} + l_{DC} \quad (12)$$

#### 3.2. Single Discriminator GAN Model with Only DSC Loss Functions (Exp\_model\_2)

Another experiment compares the model's performance with only the DSC loss function. We kept all parameters the same for this experiment, along with the same training dataset. The loss function for the update of the generator is:

$$l = l_{DSC} + 10^{-3}l_{Gen} \quad (13)$$

For the testing, we took slices from 15 to 145, 30 to 180, and 25 to 145 in axial, coronal, and sagittal planes, respectively, similar to the training data. Thus, we have 130, 150, and 120 slices per subject in the axial, coronal, and sagittal planes, respectively. We chose twenty-five subjects from the dataset which were not included in the training data. We extracted slices from each subject, calculated the average DSC score for that plane, and calculated the standard deviation for each calculated average. As discussed, the test data were predicted using three models (the proposed model and two models addressed in the experiment to check the improvement). As we aim to improve the DSC score by providing additional dissimilarity loss to the model, we compared and focused the result mainly on the DSC scores obtained by the models.

We evaluated the models using another evaluation metric, VOE, as in Table 1. From the table, it can be seen that the VOE score is decreased by up to 2% in some cases of the axial and coronal planes. The lower the value of VOE, the better the result will be. Hence, we can say that our model has less volumetric error in case of the axial and coronal planes. However, the score for the sagittal plane is similar to the Exp\_model\_1. As the values for sagittal planes are not higher than the Exp\_model\_1, and the values in two other planes are less than the Exp\_model\_1, we can claim that our proposed model outperforms the other models.

In Table 2, we compared the DSC scores of the models. We extracted 2D images from a plane and segmented it to create the segmented 3D image from those 2D slices. In the axial plane we can see that the DSC score for the WM, GM, and CSF, are higher in the proposed model by 1% than the exp\_model\_1. Also, if we compare it with the exp\_model\_2, the scores are improved by 2% for all classes. Hence, in the axial plane we can claim that our proposed model is better than other models. For the coronal plane, the scores are slightly higher in the proposed model than in the exp\_model\_1, but not by a significant margin.

On the other hand, the scores are greater than scores of exp\_model\_2 by up to 1%. The same condition occurred in the sagittal plane, where the score of the proposed model is higher than exp\_model\_2 by 1% but had just a slight improvement in comparison to the exp\_model\_1. However, we have already seen that the VOE for the axial and coronal planes is less in the proposed model. Also, in the sagittal plane, our proposed model's scores are not less than the other models. Hence, we can claim that our model outperforms other models in the segmentation of the brain tissues.

**Table 1.** VOE comparison of segmentation results from different experimental models.

Axial			
Model	CSF	GM	WM
Exp_model_1	0.0991	0.0958	0.1164
Exp_model_2	0.1075	0.1057	0.1237
Proposed model	0.0907	0.0831	0.0913
Coronal			
Exp_model_1	0.0924	0.0803	0.0688
Exp_model_2	0.0947	0.0862	0.0790
Proposed model	0.0890	0.0784	0.0688
Sagittal			
Exp_model_1	0.0885	0.0811	0.0821
Exp_model_2	0.0962	0.0938	0.1037
Proposed model	0.0877	0.0806	0.0829

**Table 2.** DSC score comparison of segmentation results of all testing subjects (average of DSC scores in total testing subjects).

Axial			
Model	CSF	GM	WM
Exp_model_1	0.9463 ± 0.0445	0.9472 ± 0.0591	0.9304 ± 0.1098
Exp_model_2	0.9399 ± 0.0703	0.9390 ± 0.0902	0.9235 ± 0.1271
Proposed model	0.9518 ± 0.0200	0.9555 ± 0.0374	0.9467 ± 0.09458
Coronal			
Exp_model_1	0.9511 ± 0.0220	0.9578 ± 0.0198	0.9635 ± 0.0312
Exp_model_2	0.9497 ± 0.0252	0.9541 ± 0.0326	0.9564 ± 0.0585
Proposed model	0.9530 ± 0.0212	0.9589 ± 0.0192	0.9634 ± 0.0340
Sagittal			
Exp_model_1	0.9534 ± 0.0198	0.9573 ± 0.0223	0.9551 ± 0.0538
Exp_model_2	0.9487 ± 0.0303	0.9492 ± 0.0457	0.9395 ± 0.0926
Proposed model	0.9537 ± 0.0199	0.9575 ± 0.0232	0.9546 ± 0.0553

#### 4. Discussion

We confirmed the superior performance of our proposed model using DSC scores in testing. We further investigate the result with or without dissimilarity coefficient by running the test with exp\_model\_1 and exp\_model\_2. The exp\_model\_1 and the proposed model have the dissimilarity coefficient as a loss function, along with the DSC. The only difference is that the model is updated once or alternatively for those loss functions. From Tables 1 and 2 it is visible that the network with the information about the dissimilarity performed better than using DSC alone. If we compare the result of the models exp\_model\_1 with dissimilarity information and exp\_model\_2 without dissimilarity information, the network's score is improved by up to 3%. In all three planes and testing subjects, the models with our proposed additional loss function outperformed the model with only the DSC loss function.

We compared our proposed method with other approaches as well, which is shown in Table 3. We want to mention that all the models listed in that table are not implemented by us. The scores shown in the table are as it was reported in the corresponding papers. The average scores of each class/tissue are the average scores of the three planes. From Table 3, we can see that our method has scored the highest DSC scores in all three tissues. If we compare with the best previous approach, as mentioned in the table, our model has performed better by 1%, 2%, and 1% in WM, GM, and CSF, respectively. The only comparable result was obtained using patch-wise models [34,35]. Therefore, we can conclude that our method has performed better than the previous state-of-the-art approaches for brain tissue segmentation.

**Table 3.** Comparison of segmentation results of different approaches.

Authors	Methods	DSC Scores		
		WM	GM	CSF
Zhang et al. [8]	CNN	86.4%	85.2%	83.5%
Nie et al. [9]	FCN	88.7%	87.3%	85.5%
Lee et al. [34]	Path-wise UNet	94.33%	93.33%	92.67%
Yamanakkanavar et al. [35]	Patch-wise Mnet	95.17%	94.32%	93.60%
	Proposed method	96.41%	96.23%	94.37%

Although the alternative update of the model is time-consuming, our proposed algorithm improved the DSC score up to 2% more than the model with DSC alone as a loss function. While comparing the single update of the model and the alternative update, we found that the alternative update outperformed in the axial and coronal planes. The scores are improved by up to 1%. However, in the sagittal plane, the performance is not significantly enhanced, but overall performance is similar to the single update. We visualized and checked the extracted slices for the sagittal plane. We found that the training and testing slices obtained are not identical as we sliced the MRI scan from a manually defined range. The slices need to be extracted dynamically and uniformly rather than defining a fixed range.

## 5. Conclusions

We have described an algorithm to update the GAN generator using the newly proposed dissimilarity coefficient, used along with DSC. The aim is to update the model to not only focus on the true positive area but also on the false positive and false negative area, which will improve the performance. Our method comprises two updates, the first one is based on the true positive, and the second one makes the correction based on the false positive and false negative area. We tested our model with the ideal condition which follows the traditional weight update. From the comparison we found that our model outperformed the others. Also, we have compared our method with the previous methods used for brain tissue segmentation. Again, in this comparison our method outperformed previous approaches. We have highlighted the limitations of a DSC loss function and introduced the dissimilarity coefficient as a new loss function. However, there are a few drawbacks of our method as it is time-consuming and resource-consuming. Despite this, in terms of the DSC scores, our model is the best one. By using segmentation in the different planes for different subjects, we have confirmed that using the dissimilarity coefficient and the alternative update of the generator is, by a considerable margin, more robust and accurate than DSC only. We evaluated and demonstrated the better performance of our model with DSC scores and the VOE scores. In the future work, we will try to study and overcome the drawbacks of our model and try to improve it to be the best in all planes.

**Author Contributions:** R.P. has developed the concept and handled the analysis. The concept has been examined by G.-R.K., and the findings have been confirmed. The paper was reviewed and contributed to by all authors, and the final version was approved by them all. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R111A3050703). And this research was supported by the BrainKorea21Four Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (4299990114316). Additionally, this work was supported by a project for Industry-University-Research Institute platform cooperation R&D funded Korea Ministry of SMEs and Startups in 2022 (S3312710).

**Institutional Review Board Statement:** The studies involving human participants were reviewed and approved by IRB No. is 2-1041055-AB-N-01-2019-45/2020-72 in Chosun University.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets utilized in this article were obtained from the OASIS webpage, which is freely accessible for all scientists and investigators to conduct experiments and can be simply accessed from OASIS's website: <https://www.oasis-brains.org/#data> (accessed on 15 March 2022).

**Acknowledgments:** Data were provided by OASIS: Cross-Sectional MRI: Principal Investigators: D. Marcus, R. Buckner, J. Csernansky, and J. Morris; P50 AG05681, P01 AG03991, P01 AG026276, R01 AG021910, P20 MH071616, U24 RR021382. Correspondence should be addressed to G.-R.K., grkwon@chosun.ac.kr.

**Conflicts of Interest:** The authors disclose that data utilized in the quantification of this study were accessed through the Open Access Series of Imaging Studies (OASIS) webpage ([oasis-brains.org](https://www.oasis-brains.org), accessed on 15 March 2022).

## References

1. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
2. Bauer, S.; Wiest, R.; Nolte, L.-P.; Reyes, M. A survey of MRI-based medical image analysis for Brain Tumor Studies. *Phys. Med. Biol.* **2013**, *58*, R97. [[CrossRef](#)] [[PubMed](#)]
3. Hsiao, C.J.; Hing, E.; Ashman, J. Trends in Electronic Health Record System Use Among Office-based Physicians: United States, 2007–2012. *Natl. Health Stat. Rep.* **2014**, *1*, 1–18.
4. Prajapati, R.; Khatri, U.; Kwon, G.R. An efficient deep neural network binary classifier for alzheimer's disease classification. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Jeju Island, Republic of Korea, 13–16 April 2021.
5. Despotović, I.; Goossens, B.; Philips, W. MRI segmentation of the Human Brain: Challenges, methods, and applications. *Comput. Math. Methods Med.* **2015**, *2015*, 450341. [[CrossRef](#)]
6. Duta, N.; Sonka, M. Segmentation and interpretation of mr brain images. an improved active shape model. *IEEE Trans. Med. Imaging* **1998**, *17*, 1049–1062. [[CrossRef](#)] [[PubMed](#)]
7. Fischl, B.; Salat, D.H.; Busa, E.; Albert, M.; Dieterich, M.; Haselgrove, C.; van der Kouwe, A.; Killiany, R.; Kennedy, D.; Klaveness, S.; et al. Whole brain segmentation. *Neuron* **2002**, *33*, 341–355. [[CrossRef](#)]
8. Zhang, W.; Li, R.; Deng, H.; Wang, L.; Lin, W.; Ji, S.; Shen, D. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage* **2015**, *108*, 214–224. [[CrossRef](#)] [[PubMed](#)]
9. Nie, D.; Wang, L.; Gao, Y.; Shen, D. Fully convolutional networks for multi-modality isointense infant brain image segmentation. In Proceedings of the 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), Prague, Czech Republic, 13–16 April 2016.
10. Shattuck, D.W.; Sandor-Leahy, S.R.; Schaper, K.A.; Rottenberg, D.A.; Leahy, R.M. Magnetic resonance image tissue classification using a partial volume model. *NeuroImage* **2001**, *13*, 856–876. [[CrossRef](#)] [[PubMed](#)]
11. Wells, W.M.; Grimson, W.E.L.; Kikinis, R.; Jolesz, F.A. Adaptive segmentation of MRI Data. *IEEE Trans. Med. Imaging* **1996**, *15*, 429–442. [[CrossRef](#)] [[PubMed](#)]
12. Xue, H.; Srinivasan, L.; Jiang, S.; Rutherford, M.; Edwards, A.D.; Rueckert, D.; Hajnal, J.V. Automatic segmentation and reconstruction of the cortex from neonatal MRI. *NeuroImage* **2007**, *38*, 461–477. [[CrossRef](#)]
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 139–144.
14. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. *Adv. Neural Inf. Process. Syst.* **2015**, *1*, 1486–1494.

15. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
16. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *arXiv* **2014**, arXiv:1606.03498.
17. Zhao, J.; Mathieu, M.; LeCun, Y. Energy-based generative adversarial network. *arXiv* **2016**, arXiv:1609.03126.
18. Vondrick, C.; Pirsivash, H.; Torralba, A. Generating videos with scene dynamics. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 613–621.
19. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
20. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative adversarial text to image synthesis. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1060–1069.
21. Zhao, Z.; Wang, Y.; Liu, K.; Yang, H.; Sun, Q.; Qiao, H. Semantic segmentation by improved generative adversarial networks. *arXiv* **2021**, arXiv:2104.09917.
22. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
23. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
24. Hardy, C.; Le Merrer, E.; Sericola, B. MD-Gan: Multi-discriminator generative adversarial networks for distributed datasets. In Proceedings of the 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Rio de Janeiro, Brazil, 20–24 May 2019.
25. Dice, L.R. Measures of the amount of ecologic association between species. *Ecology* **1945**, *26*, 297–302. [[CrossRef](#)]
26. Prajapati, R.; Kwon, G.-R. SIP-UNet: Sequential Inputs Parallel UNet Architecture for Segmentation of Brain Tissues from Magnetic Resonance Images. *Mathematics* **2022**, *10*, 2755. [[CrossRef](#)]
27. Marcus, D.S.; Wang, T.H.; Parker, J.; Csernansky, J.G.; Morris, J.C.; Buckner, R.L. Open access series of imaging studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J. Cogn. Neurosci.* **2007**, *19*, 1498–1507. [[CrossRef](#)] [[PubMed](#)]
28. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Kong, Y.; Li, H.; Ren, Y.; Genchev, G.Z.; Wang, X.; Zhao, H.; Xie, Z.; Lu, H. Automated yeast cells segmentation and counting using a parallel U-Net based two-stage framework. *OSA Contin.* **2020**, *3*, 982. [[CrossRef](#)]
33. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
34. Lee, B.; Yamanakkanavar, N.; Choi, J. Automatic segmentation of brain MRI using a novel patch-wise U-net deep architecture. *PLoS ONE* **2020**, *15*, e0236493. [[CrossRef](#)]
35. Yamanakkanavar, N.; Lee, B. Brain Tissue Segmentation using Patch-wise M-net Convolutional Neural Network. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics—Asia (ICCE-Asia), Seoul, Korea, 1–3 November 2020; pp. 1–4.