DOI: 10.6688/JISE.202409_40(5).0010

# Semi-Supervised Learning Using Co-Generative Adversarial Network (Co-GAN) for Medical Image Segmentation

GUO-QIN LI[1,2], NURSURIATI JAMIL[2] AND RASEEDA HAMZAH[3,+]
[1]*Taiyuan Institute of Technology*
*Taiyuan, 030008 P.R. China*
[2]*College of Computing, Informatics and Media*
*Universiti Teknologi MARA (UiTM)*
*Selangor, 40450 Malaysia*
[3]*College of Computing, Informatics and Media*
*Universiti Teknologi MARA (UiTM), Melaka Branch*
*Merlimau, Melaka, 77300 Malaysia*
*E-mail: raseeda@uitm.edu.my*[+]

Medical image analysis has experienced different stages of development, especially with the emergence of deep learning. However, acquiring large-scale, high-quality labeled data to train a deep learning model takes time and effort. This paper proposes a semi-supervised learning method for medical image segmentation using limited labeled data and large-scale unlabeled data. Inspired by the classic Generative Adversarial Network (GAN) and co-training strategy, we proposed a new Co-GAN framework to implement medical image segmentation. The proposed Co-GAN comprises two generators and one discriminator, in which two generators can provide mutual segmentation information to each other. Through adversarial training between generators and discriminators, Co-GAN achieved higher segmentation accuracy. The dataset used was the hippocampus in Medical Segmentation Decathlon (MSD). There were four training data settings: 25 labeled slices/3,374 unlabeled slices; 50 labeled slices/3,349 unlabeled slices; 100 labeled slices/3,299 unlabeled slices; and 200 labeled slices/3,199 unlabeled slices. Three experiments were conducted for each data set: fully supervised learning based on a generator network using only labeled data (F-Generator), semi-supervised learning based on GAN (Semi-GAN), and semi-supervised learning based on Co-GAN. The experiments showed that Co-GAN improved the segmentation accuracy by (1.9%, 2.6%, 1.1%, and 0.1%) compared to F-Generator and (2.2%, 0.8%, 0.5%, 0.7%) to Semi-GAN.

***Keywords:*** semi-supervised learning, GAN, co-training, Co-GAN, medical image segmentation

## 1. INTRODUCTION

Deep learning algorithms such as Convolutional Neural Networks have made rapid development in medical image segmentation, as reported by [1-7]. Most segmentation models are trained to extract features using fully-supervised learning, which requires massive labeled data for training to get satisfactory segmentation results. It is challenging to acquire large-scale and carefully labeled datasets for medical images. The process of obtaining a perfect dataset is rather time-consuming and labor-intensive, which leads to the expense of data collection. Meanwhile, unlabeled data are more accessible than labeled data, and many unlabeled data are produced daily in clinical practice. Thus, many researchers have turned to semi-supervised learning studies for medical image segmentation, as

Received February 17, 2023; revised May 24 & July 4 & July 24, 2023; accepted August 17, 2023.
Communicated by Lee Hung Liew.

[+] Corresponding author.

done by [8-12], aiming to alleviate the burden of collecting labeled data. In semi-supervised learning-based segmentation methods, the information of unlabeled data is extracted to train the model along with limited labeled data. For example, the authors in [10] combined post-processing with self-training to achieve segmentation of ventricular MRI. This method would train the segmentation network with labeled data, and then unlabeled data was fed into the network to get a segmentation map. Consequently, the segmentation map was optimized by post-processing. The refined map was directly taken as the additional ground truth for updating the network parameters. In [11], the authors improved the training strategy of [10] by using a reasonable part of the region of the segmentation map of the unlabeled data and combined it with the labeled data to improve the training procedure. It shows that an unlabeled data segmentation map (also called the predicted map) is crucial in the training process. Thus, it is vital to improve the predicted map's quality for unlabeled data to achieve higher segmentation accuracy.

One of the most popular approaches in semi-supervised learning is called co-training, initially proposed by Blum and Mitchell [13]. Co-training trains multiple models by first using labeled data, in which each model should be trained sufficiently by a single view of the data. Then, co-training minimizes the disagreements by assigning pseudo labels between each view on unlabeled data [14]. In the co-training process, the predicted maps from different perspectives should be agreed upon for the same unlabeled data, enforcing multiple models to be generalized well to the unlabeled data [15]. With the development of the co-training approach, the idea of co-training has been applied to different research studies, such as data augmentation [16-17], image segmentation [14, 15, 18-20], image classification [21], and image recognition [22-24].

For image segmentation, the authors in [14] proposed an uncertainty-aware multi-view co-training (UMCT) framework to segment 3D medical images in which an uncertainty-weighted label fusion mechanism was proposed to estimate the reliability of the predicted map for each view. In [15], the authors proposed a deep adversarial co-training method for 2D medical image semantic segmentation. They used multiple models to predict the pseudo labels and fused them to get the average. At the same time, adversarial samples were introduced to capture the difference between the models to make the models learn more complementary information in the training process.

Meanwhile, the authors in [18] proposed a deep attention network to find and correct erroneous information in noise labels adaptively and consequently proposed a hierarchical distillation method to generate more reliable predicted maps. When developing a predicted map, each model's prediction in model distillation was replaced by the prediction of each model under multiple data transformations. Then the predictions were fused to get the final predicted map, which further improves the quality of the predicted map. In [19], the authors proposed SMU-Net to achieve brain tumor segmentation with missing modalities. The authors utilized a co-training strategy to distill the useful information from the full-modality path into a missing modality path, in which co-training encourages the missing modality network to reconstruct the missing information. In [20], the authors proposed a deep multiplanar co-training (DMPCT) to achieve abdominal multi-organ segmentation. The co-training could mine consensus information from multiple planes like sagittal, coronal, and axial planes, and then multi-planar fusion was applied to generate more reliable predicted maps. In the co-training approach used by the above-mentioned studies, the mean or weighted fusion of the predicted maps may affect the training results. Although the fusion of predicted maps can improve training performance, the measurement to accurately estimate the confidence of the predicted maps is still an essential problem [25]. Several predicted maps with low segmentation performance during fusion will affect the training outcome.

Recently, generative adversarial network (GAN) [26] based approaches are also popular for medical image processing, and some prominent extension models were also proposed in [8, 9, 27-34]. GAN consists of two networks: generator and discriminator. These two networks were in an adversarial two-players game [28] in which a generator was used to generate a predicted map with high similarity to the ground truth of the real data, whereas a discriminator was used to estimate whether the predicted map came from the ground truth of real data or outputs of the generator.

For example, the authors in [30] used GAN to generate predicted samples for unlabeled data to enlarge the training dataset size, in which the ability of the discriminator will significantly affect the performance of the entire adversarial learning process. In [8], the authors proposed Deep Adversarial Network (DAN) to perform adrenal and fungal segmentation tasks. The researchers designed an adversarial loss function that used unlabeled data to control the training process better. In [28], the authors refined the goal of the discriminator, which can output a confidence map of the predicted map. The predicted map with high confidence was chosen as the pseudo-label map to train the model.

Consequently, the entire model was iteratively optimized by the self-training method. In [33], the authors used the discriminator to distinguish the quality of predicted maps from the segmentation network. At the same time, a classification network was utilized to assist in optimizing the predicted maps by using attention features. In [34], the authors proposed a GAN-based model for Breast Ultrasound Image (BUS) image segmentation. A dual-attentive-fusion block was designed to extract features of the lesion region and background separately, which makes more effective discrimination for the quality of predicted maps. Besides, many medical image segmentation approaches based on the extension of GAN have been studied, as described in the paper [27]. Based on this literature on extended GANs, one generator commonly corresponds to one discriminator to achieve medical image segmentation.

The adversarial training between GAN's generator and discriminator shows that a generator can change the usage of an unlabeled dataset. The training procedure yields a more trustworthy predicted map that can deceive the discriminator into believing that the predicted map has a high degree of similarity to the ground truth. Thus, the whole performance of GAN is improved to produce final satisfactory segmentation results. Conversely, the co-training strategy can use the predicted maps of different data views to supervise each other in multiple models. The strengths of adversarial training of GAN and co-training design are complementary and are the aim of this paper. Thus, the combination of a co-training strategy and GAN is proposed and named Co-GAN to produce high-quality predicted maps for medical image segmentation. In Co-GAN, two generators and one discriminator were adopted compared to extended GANs [27]. The two generators utilize the co-training strategy to provide mutual segmentation information.

Furthermore, the adversarial training between two generators and one discriminator allows better interaction among each other. One generator corresponds to one discriminator in each co-training stream [15]. There needs to be more mutual guidance between these streams. In Co-GAN, generators can extract more feature information, conserving computational space, despite just one utilization of one discriminator. Thus, Co-GAN significantly improved the quality of predicted maps.

This paper concludes the significant contributions as follows:

(1) A semi-supervised learning based on co-training and GAN (Co-GAN) was proposed to use information from the unlabeled data fully. The proposed method can improve image segmentation while only using limited labeled data and large-scale unlabeled data.
(2) In Co-GAN, there were two generators and one discriminator. Because the unlabeled data lacks ground truth, the two generators can provide mutual segmentation information to each other. The discriminator was used to distinguish between the predicted map produced by the generator and the ground truth. Through adversarial training, Co-GAN can achieve higher segmentation accuracy because the discriminator can guide the generator to generate a more accurate predicted map. The mutual information guidance of the two generators can also improve segmentation performance.
(3) According to the experiment results, the proposed Co-GAN can achieve more accurate segmentation results. Co-GAN can alleviate the burden of clinical experts for the annotation process and improve the utilization of unlabeled data.

## 2. METHOD

This paper compares fully supervised (F-generator), semi-supervised (semi-GAN), and the proposed Co-GAN framework to understand their performance in leveraging largescale unlabeled data for medical image segmentation. Figs. 1-3 introduce the architecture of Co-GAN and two comparison models. The encoder-decoder structure is one of the most popular architectures for medical applications. Thus, all three models utilized an encoder-decoder deep learning network known as a generator. F-Generator only used labeled data to perform segmentation, as shown in Fig. 1. The second model, semi-GAN (see Fig. 2), used a generator and discriminator for the segmentation. Semi-GAN utilized both labeled and unlabeled data. Fig. 3 depicts the proposed Co-GAN, which consists of two generators and a discriminator. In Co-GAN, the interaction between the generators and discriminator enhances segmentation performance. The discriminators of semi-GAN and Co-GAN also adopted the same network structure. The subsequent section describes the details of the F-generator, semi-GAN, and Co-GAN.

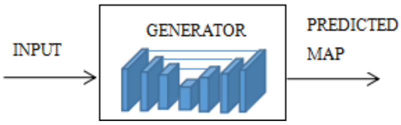

Fig. 1. F-generator (input contains labeled data only).

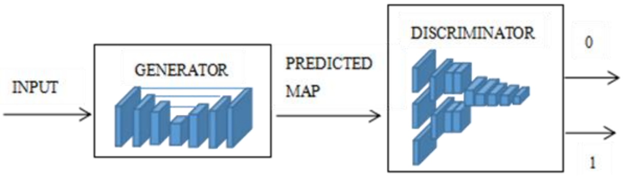

Fig. 2. Semi-GAN (input contains labeled data and unlabeled data).

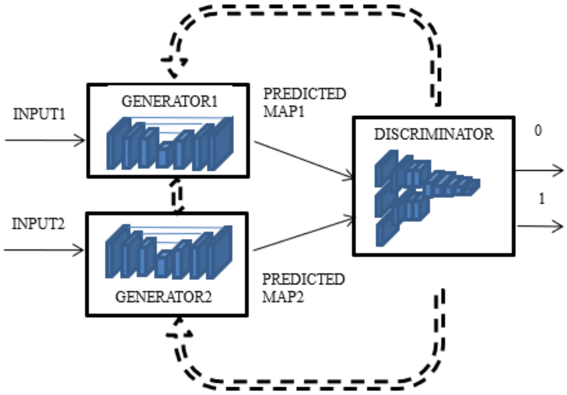

Fig. 3. Co-GAN (input contains labeled data and unlabeled data, and dotted lines represents main interactions).

### 2.1 Overview of the Co-GAN Framework

Fig. 4 shows the proposed Co-GAN framework consisting of two generators and one discriminator. Generator 1 receives labeled data (original image and ground truth) and unlabeled data (original image) to generate corresponding predicted maps. Meanwhile, Generator 2 accepts rotated labeled data (rotated original image and rotated ground truth) and rotated unlabeled data (rotated original image) to generate corresponding rotated predicted maps. Conversely, the discriminator acquires five resources: predicted map with the original image for labeled data (predicted map 1), predicted map with the original image for unlabeled data (predicted map 2), ground truth with the original image for labeled data, rotated predicted map with the rotated original image for rotated labeled data (rotated predicted map 1), and rotated predicted map with the rotated original image for rotated unlabeled data (rotated predicted map 2).

In GAN, the generator generates a predicted map closer to the ground truth, while the discriminator distinguishes between the predicted map and the ground truth. The discriminator outputs an evaluation score for each input data *i.e.* 0 and 1. The input data to the discriminator which is the ground truth, was represented with 1, while 0 represents the predicted map's input data. Therefore, the closer the score was to 1, the more realistic the input of the discriminator is, which means the closer to the ground truth. However, the closer the score was to 0, the more fictitious the information of the discriminator was, which means the closer to the predicted map, and the quality was too poor. The continuous iterative optimization improved the adversarial training process.

In Co-GAN, there were two generators available. Aside from the adversarial training process between generators and discriminators, the two generators in Co-GAN have another goal: to provide mutual guidance for each other. The predicted map from Generator 1 and the rotated predicted map from Generator 2 should reach a consensus for one data. Hence, the expected map from each generator can be taken as fake ground truth to supervised one another to learn the mutual information.

In Fig. 4, similarity loss was the loss function to evaluate the difference between predicted maps from the two generators. The smaller the similarity loss value was, the more similar the predicted maps are, which means two generators could reach an agreement for one data. Since Co-GAN only has one discriminator and can guide the generator optimization, Generator 1 and Generator 2 can influence each other through adversarial training and mutual guidance from the similarity loss.

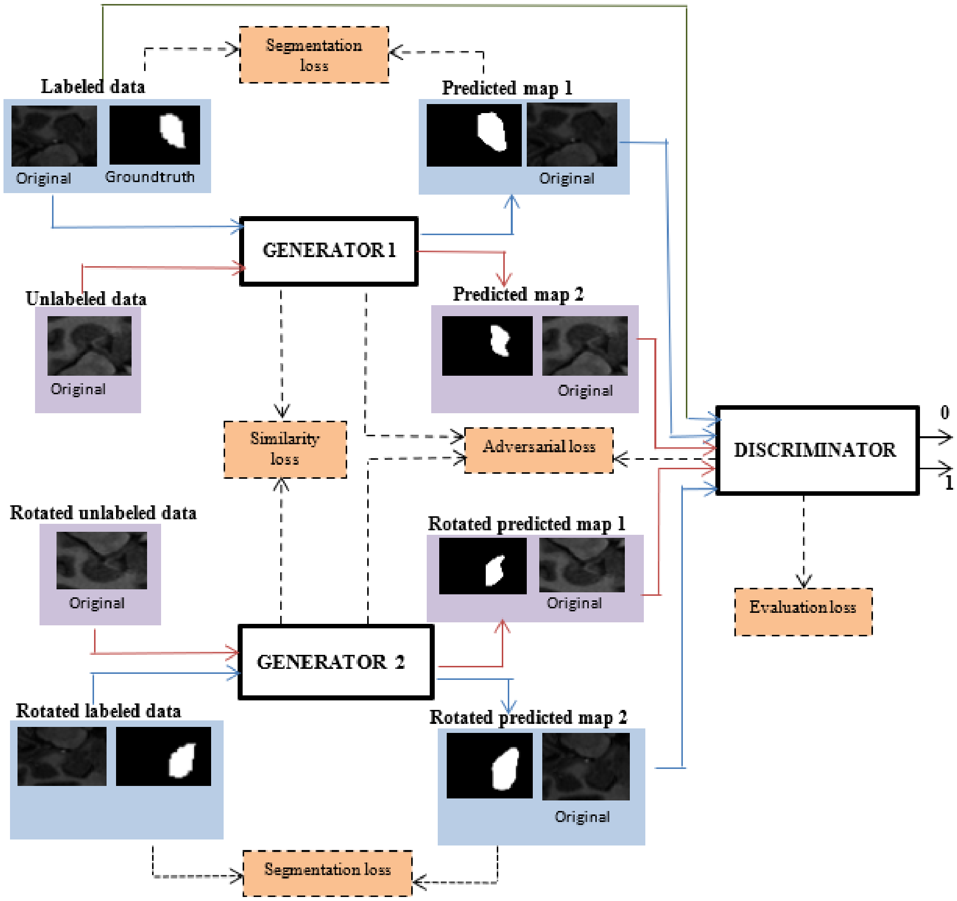

Fig. 4. The detailed architecture of the proposed Co-GAN framework.

(A) Design of generators

In Co-GAN, two generators are designed to generate predicted maps for input data, that is medical image in this paper. The interaction between two generators in this paper reflects the co-training strategy. Thus, there were two data views required for co-training. One data view was the original image, and the other was the rotated original image in which the rotation angle is 180°. The 180° chosen is easy to implement in experiments, and also has suitable performance for enlarging the difference of views for co-training strategy. Generator 1 accepts original images as input, while Generator 2 accepts rotated original images. Nevertheless, the networks of the two generators have the same structure, as shown in Fig. 5.

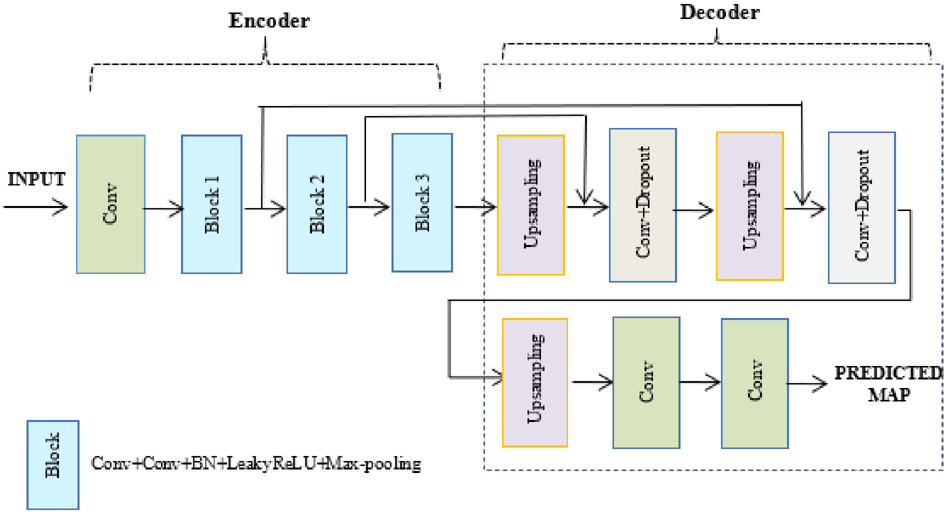

Fig. 5. The encoder-decoder structure of the generator.

The network of the generator was an encoder-decoder structure. One block unit contains two convolution layers with kernel size 3×3, batch normalization (BN) layer, one LeakyReLU layer, and one Max-pooling layer with pool_size 2. For each generator in Co-GAN, the input data contains labeled and unlabeled data. Firstly, a convolution with kernel size 1×1 was applied to the input data and transferred into three consecutive block units, and the feature maps for convolution layer in each block unit are 64, 128, 256 respectively. Hence, the three-block units can fully extract the low-level and high-level features from the input data of size 40×56. The rest of the network was the decoder, consisting of three upsampling layers. A convolutional with kernel size 3×3 + dropout layer was added between the two upsampling layers because the feature map needs to be restored to a specific size with high-level feature information. It should be noted that the feature maps with the exact resolution were concatenated together between the encoder and decoder to combine low-level and high-level information to improve segmentation accuracy. Finally, two convolution layers with kernel size 3×3 and 1×1 respectively after the final upsampling were used to merge the channel of the feature maps into one to get the predicted map.

In the Co-GAN training process, the similarity loss between two generators was used to measure the similarity of predicted maps for one input data. Thus, the predicted map from Generator 2 must be rotated 180° again when calculating the similarity because generator 2 accepts rotated labeled data (rotated original image and rotated ground truth). The rotated unlabeled data (rotated original image) generates corresponding rotated predicted maps.

(B) Design of discriminator

The discriminator aims to distinguish between the predicted map and the ground truth. An evaluation score between 0 to 1 was given to the input data representing the similarity of the input to either the predicted map or the ground truth. The closer the score was to 1, the more realistic the input data was, which means the closer to the ground truth. However, if the score was closer to 0, the more fictitious the input data was, which means the closer to the predicted map.

In Co-GAN, the discriminator has five input data resources: predicted map with the original image for labeled data, predicted map with the original image for unlabeled data, ground truth with the original image for labeled data, rotated predicted map with the rotated original image for rotated labeled data, and rotated predicted map with the rotated original image for rotated unlabeled data. For the sake of the description, it should be clear that the predicted map and ground truth both represent the region-of-interest (ROI) segmentation of the original image. The difference between them was that the predicted map was from the generator, while the ground truth was from experts' annotation. When training the discrimination, there were three channels for one input data. The first channel was the original image, the second channel was the corresponding segmented ROI (predicted map or ground truth), and the third channel was the inverse segmented ROI (inverse predicted map or inverse ground truth).

The architecture of the discrimination was shown in Fig. 6. First, the input for each channel was convolved twice, using a kernel size of 1×1, and kernel size of 3×3. Then the original image and its segmented ROI (ground truth) were multiplied to produce Output 1. The original image and the inverse segmented ROI were simultaneously multiplied to generate Output 2. Thus, Output 1 provides information on ROI, and Output 2 provides information on the background. Next, the outputs implement the block operations twice (each block contains convolution + batch normalization + LeakyReLU). Consequently, the outputs after the operations of the block were concatenated to create Output 3. Two block operations were applied on Output 3 to produce Output 4. Finally, Output 4 was subjected to a 1×1 convolution, flatten, and dense to get the evaluation score to distinguish the source of input data.

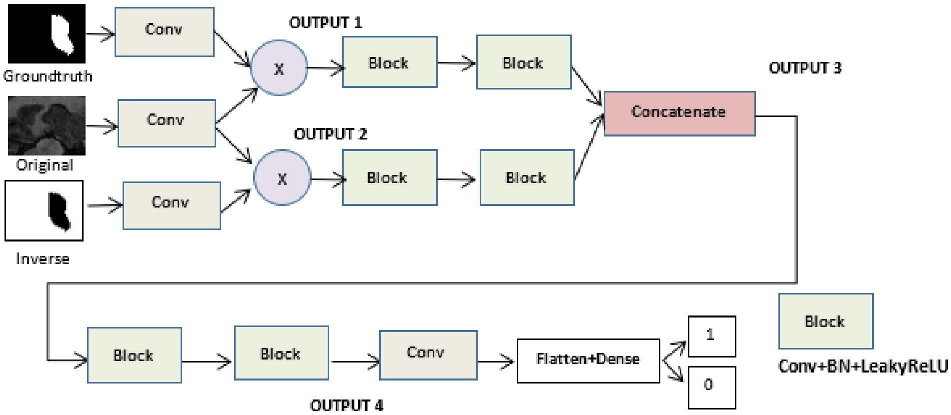

Fig. 6. The architecture of the discriminator.

## 2.2 Training Strategy of Semi-supervised Learning based on Co-GAN Framework

In this section, the training strategy in Co-GAN consists of a semi-supervised learning process and the loss function definition. First, the semi-supervised learning process was introduced, and then the loss function definition in the training process was described in detail.

(A) Semi-supervised learning process

Fig. 4 shows that there were two generators and one discriminator in Co-GAN. For each generator, the input was a single view of the training data. In this paper, one view of the training data was the original image, and the other view of the training data was the rotated original image in which the rotation angle is 180°. For the initialization process, each generator was trained by using only the labeled data. This was a fully-supervised learning. However, the two generators and one discriminator are combined to represent the proposed Co-GAN. The Co-GAN framework used semi-supervised learning dataset comprising unlabeled and limited labeled data.

(B) Loss function definition

The loss function provides the direction of model optimization, which was utilized to calculate the predicted error during the training process. The predicted error presents the difference between the ground truth and the predicted map. Then, the model was trained to minimize the predicted error.

The first step in Co-GAN strategy was to initialize the two generators by training them with labeled data. The segmentation loss function for Generator 1 was notated by $\boldsymbol{loss_{seg1}}$ and defined as follows,

$$\boldsymbol{loss_{seg1}} = \boldsymbol{loss_{dice1}} + \boldsymbol{loss_{dis1}} \tag{1}$$

where

$$loss_{dice1} = 1 - Dice(Y_L^1, \bar{Y}_L^1) = 1 - 2\frac{|Y_L^1 \cap \bar{Y}_L^1|}{|Y_L^1| + |\bar{Y}_L^1|} \tag{2}$$

and

$$loss_{dis2} = -Y_L^1 \cdot \log \bar{Y}_L^1 - (1 - Y_L^1) \cdot \log(1 - \bar{Y}_L^1) \tag{3}$$

$Y_L^1$ = ground truth of the labeled data used by Generator 1
$\bar{Y}_L^1$ = predicted map of the labeled data generated by Generator 1

Similarly, the loss function for Generator 2 notated by $\boldsymbol{loss_{seg2}}$ was defined as follows,

$$\boldsymbol{loss_{seg2}} = \boldsymbol{loss_{dice2}} + \boldsymbol{loss_{dis2}} \tag{4}$$

where

$$loss_{dice1} = 1 - Dice(Y_L^2, \bar{Y}_L^2) = 1 - 2\frac{|Y_L^2 \cap \bar{Y}_L^2|}{|Y_L^2| + |\bar{Y}_L^2|} \tag{5}$$

and

$$loss_{dis2} = -Y_L^2 \cdot \log(\bar{Y}_L^2) - (1 - Y_L^2) \cdot \log(1 - \bar{Y}_L^2) \tag{6}$$

where

$Y_L^2$ = ground truth of the labeled data used by Generator 2
$\bar{Y}_L^2$ = predicted map of the labeled data generated by Generator 2

In general, the segmentation loss notated by $loss_{seg}$ represents the supervised segmentation loss between the predicted map and the ground truth and only works with labeled data, measuring how similar they are. Meanwhile, $loss_{dis}$ was the binary cross-entropy loss. These two losses simultaneously control the initialization of generators. The optimizer used in the initialization process was Adam, while the training epochs were set to 150 and the batch size was set to 16.

The second step was to train the combination of two generators and one discriminator using unlabeled data and limited labeled data. The loss functions in this step were defined in the generator and discriminator optimization process. Eq. (7) defines the generator's loss function in which $\alpha$ and $\beta$ denote $loss_{simi}$ and $loss_{adv}$ weight. The derivations of each loss weight were presented in Eqs. (8)-(10).

$$loss_G = loss_{seg} + \alpha loss_{dis2} + \beta loss_{adv} \tag{7}$$

$$loss_{seg} = loss_{seg1} + loss_{seg2} \tag{8}$$

$$loss_{simi} = l_{bce}(\bar{Y}_L^1, \bar{Y}_L^2) + l_{bce}(\bar{Y}_U^1, \bar{Y}_U^2) \tag{9}$$

$$\begin{aligned} loss_{adv} = {} & l_{bce}(D(X_L^1, \bar{Y}_L^1), 1) + l_{bce}(D(X_U^1, \bar{Y}_U^1), 1) \\ & + l_{bce}(D(X_L^2, \bar{Y}_L^2), 1) + l_{bce}(D(X_U^2, \bar{Y}_U^2), 1) \end{aligned} \tag{10}$$

where

$X_L^1$ = labeled data used by Generator 1
$X_U^1$ = unlabeled data used by Generator 1
$X_L^2$ = labeled data used by Generator 2
$X_U^2$ = unlabeled data used by Generator 2
$\bar{Y}_L^1$ = predicted map for labeled data generated by Generator 1
$\bar{Y}_L^1$ = predicted map for unlabeled data generated by Generator 1
$\bar{Y}_U^2$ = predicted map for labeled data generated by Generator 2
$\bar{Y}_U^2$ = predicted map for unlabeled data generated by Generator 2

In the semi-supervised learning process, labeled data was used to train two generators, preventing large errors in the Co-GAN training process when using unlabeled data. The similarity loss notated by $loss_{simi}$ describes the similarity of the predicted maps from two generators for one data and $l_{bce}(\cdot)$ was the binary cross-entropy loss. It should be noted that the predicted map from Generator 2 was rotated compared to Generator 1. When calculating $loss_{simi}$, $\bar{Y}_L^2$ and $\bar{Y}_U^2$, the predicted map should be rotated 180° again. According to the $loss_{simi}$ function definition, two generators can provide segmentation information to each other so that the performance of Co-GAN can be improved by mutual guidance between the two generators.

Meanwhile, the adversarial loss notated by $loss_{adv}$ represents adversarial loss between the generator and discriminator. Additionally, $loss_{adv}$ is minimized by using labeled and unlabeled data. The adversarial loss was designed to estimate the degree of the predicted map, which appears similar to the ground truth. The generator aims to generate the predicted map and deceiving the discriminator into thinking that the predicted map was the ground truth. Meanwhile, $l_{bce}$ refers to the binary cross-entropy function. $\alpha$ was the weight of $loss_{simi}$ and $\beta$ was the weight of $loss_{adv}$, both of which were set to 0.02 according to the experiments training process in this paper, ensuring the stable training of Co-GAN under different training data settings.

For discriminator optimization, the aim was to make the correct distinction between the predicted map and the ground truth. In Co-GAN, the discriminator has five input data resources: predicted map with the original image for labeled data, predicted map with the original image for unlabeled data, ground truth with the original image for labeled data, rotated predicted map with the rotated original image for rotated labeled data, and rotated predicted map with the rotated original image for rotated unlabeled data. The discriminator gives a score for input data in which the standard score is 1 and 0. The input data of the discriminator from the ground truth was represented with 1, while 0 represents the input data of the discriminator from the predicted map. The closer the score was to 1, the more realistic the input data was, which means the closer to the ground truth. Conversely, the closer the score was to 0, the more fictitious the input data was, which means the closer to the predicted map.

During the adversarial training process, the discriminator should make the right evaluation for distinguishing between the predicted map and the ground truth. Then, the discriminator can provide the right guidance for optimization of the generator. The evaluation loss functions denoted by $loss_{eva}$ were defined as follows,

$$loss_{eva} = loss_{eva1} + loss_{eva2} \tag{11}$$

$$loss_{eva1} = l_{bce}(D(X_L^1, \bar{Y}_L^1), 0) + l_{bce}(D(X_U^1, \bar{Y}_U^1), 0) + l_{bce}(D(X_L^1, Y_L^1), 1) \tag{12}$$

$$loss_{eva2} = l_{bce}(D(X_L^2, \bar{Y}_L^2), 0) + l_{bce}(D(X_U^2, \bar{Y}_U^2), 0) \tag{13}$$

where
$X_L^1$ = labeled data used by Generator 1
$X_U^1$ = unlabeled data used by Generator 1
$X_L^2$ = labeled data used by Generator 2
$X_U^2$ = unlabeled data used by Generator 2
$\bar{Y}_L^1$ = predicted map for labeled data generated by Generator 1
$\bar{Y}_L^1$ = predicted map for unlabeled data generated by Generator 1
$\bar{Y}_U^2$ = predicted map for labeled data generated by Generator 2
$\bar{Y}_U^2$ = predicted map for unlabeled data generated by Generator 2
$l_{bce}$ = binary cross-entropy function
$loss_{eva1}$ = evaluation loss of Generator 1
$loss_{eva2}$ = evaluation loss of Generator 2

In the training process, the optimizer for the generator was SGD, and the learning rate was set to 0.01. On the other hand, for the discriminator, the optimizer was RMSprop, and the learning rate was set to 0.001. The batch_size of the labeled data was set to 16, and the batch_size of the unlabeled data was set to 48. During optimization, the discriminator was trained to minimize $loss_{eva}$, and generators were trained to minimize $loss_G$. The generators use labeled and unlabeled data to first minimized $loss_{simi}$ and $loss_{adv}$, and then $loss_{seg}$ was optimized using only labeled data. In the training process, the generator was trained ten times, while the discriminator was trained once. Thus, the Co-GAN framework was stable as a trained generative model.

## 3. EXPERIMENTS RESULTS

This section introduces the dataset, experimental realization, and result analysis. Overall, the section starts with the dataset description and the pre-processing steps followed by the experimental execution process. Lastly, the section presents the discussion on the result to highlight the importance of all the experiments conducted.

### 3.1 Dataset

The dataset used in this research was the hippocampus images in the Medical Segmentation Decathlon (MSD) dataset originated from Vanderbilt University Medical Centre. The hippocampus dataset contains 260 3D volumes of the MRI modality. In this paper, the target area was the anterior hippocampus. Fig. 7 shows one anterior hippocampus image viewed in three planes: Axial, Sagittal, and Coronal using 3D slicer software.

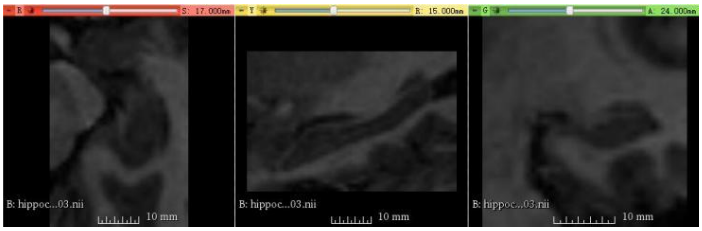

Fig. 7. Hippocampus data (Axial, Sagittal, and Coronal plane views from left to right).

Since the proposed Co-GAN framework is a 2D network, the MRI volumes and ground truth need to be converted to 2D images. In each direction, MRI data contains several 2D slices. In this paper, the 2D images were from the 2D axial slices. For the hippocampus dataset, each data has 29–40 slices in the axial plane. During the conversion, the slices without the target region were discarded because these slices cannot provide the information of ROI. Then, all 2D slices were separated into the training and testing datasets. The training dataset contains 3,399 slices with 2D ground truth, while the testing dataset contains 300 slices with 2D ground truth.

After completing 3D to 2D conversion, the images were resized, and the grey values were normalized to (0,1). The size of the original images was 35×51, which was resized to 40×56 to ensure the feature maps were of the same size in concatenation operation between encoder and decoder.

For semi-supervised learning based on Co-GAN, the training dataset needs to be further separated into either a labeled or an unlabeled dataset. The labeled dataset consists of slices and ground truth, while the unlabeled data contains only the slices, and the corresponding ground truth was discarded. When the labeled and unlabeled datasets were successfully created, the two views of the dataset need to be constructed. In this paper, the labeled and unlabeled datasets were fed into Generator 1, whereas rotated labeled and rotated unlabeled datasets are fed into Generator 2. Fig. 8 shows the dataset processing operations in this paper.

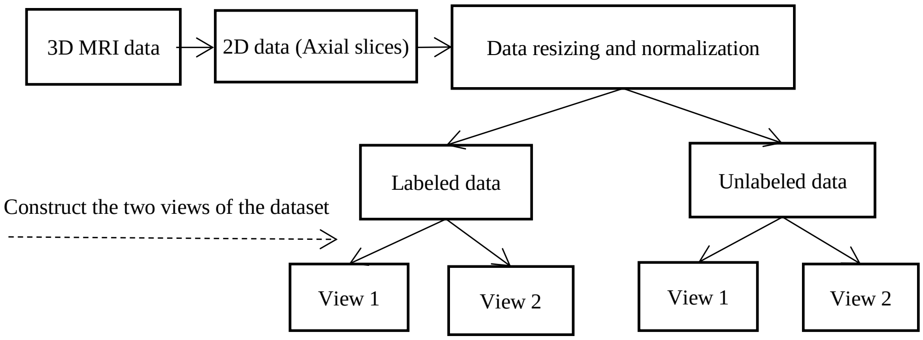

Fig. 8. Dataset pre-processing.

### 3.2 The Experiments

The proposed semi-supervised Co-GAN framework was implemented on Keras deep learning API and all the experiments were conducted on Kaggle platform.

In this paper, there are four training data settings for the number of labeled and unlabeled data. For each data setting, we conducted three experiments: fully supervised learning based on a generator network using only labeled data (F-Generator), semi-supervised learning based on GAN, and semi-supervised learning based on Co-GAN as illustrated in Table 1. Experiment One used four different number of labeled slices for training that are 25, 50, 100 and 200 slices, respectively. Since Experiment Two and Three were semisupervised learning, both the labeled and unlabeled slices were used. The same number of labeled slices used in Experiment One was also utilized in the training of Experiment Two and Three. The unlabeled slices used in Experiment Two and Three to train the respective semi-supervised GAN and Co-GAN were 3,374, 3,349, 3,299 and 3,199 slices.

**Table 1. Experiments process setting.**

| Experiment | Learning model | Dataset settings | | | | |
|---|---|---|---|---|---|---|
| One | Fully-supervised Generator (F-Generator) | Labeled slices | 25 | 50 | 100 | 200 |
| Two | Semi-supervised GAN (Semi-GAN) | Labeled slices | 25 | 50 | 100 | 200 |
| | | Unlabeled slices | 3,374 | 3,349 | 3,299 | 3,199 |
| Three | Semi-supervised Co-GAN | Labeled slices | 25 | 50 | 100 | 200 |
| | | Unlabeled slices | 3,374 | 3,349 | 3,299 | 3,199 |

(A) F-Generator

The aim of Experiment One was to initialize the two generators in Co-GAN framework using fully supervised learning. The training dataset for fully supervised learning based on generator network used only labeled data as stated in Table 1. The learning rate in the training process adopted ReduceLROnPlateau() in Keras. When the monitored loss value remains unchanged, ReduceLROnPlateau() reduced the learning rate to get better results. In the fully-supervised learning, the epoch was set to 150, and the batch_size set to 16. The experiments also showed that the loss function value of training was not changed after 150 epochs. Generator 1 initialization training was referred to as *Seg1only*. Likewise, Generator 2 initialization training was called *Seg2only* for short.

(B) Semi-GAN

The training dataset for semi-supervised learning based on GAN (Semi-GAN) contains both labeled and unlabeled data. The training data settings included 25/3,374, 50/ 3,349, 100/3,299, and 200/3,199 notated as labeled slices/unlabeled slices. This experiment aimed to justify the effectiveness of unlabeled data utilization in the training process. GAN was adopted to achieve adversarial training between generator and discriminator to ensure the quality of the predicted map of the generator. In this experiment, the generator was first initialized and then combined with the discriminator to be trained using the unlabeled data and limited labeled data. The networks of the generator and discriminator were equivalent to Co-GAN framework. For a fair comparison to F-Generator model, the initialized generator in Semi-GAN adopts *seg1only*, which means only one view of the training dataset was used to justify the effectiveness of GAN. In this experiment, the learning rate of the generator was set to 0.01 and the optimizer used was stochastic gradient descent (SGD). Meanwhile, the discriminator used a learning rate of 0.001, and the optimizer was root mean squared propagation (RMSprop). In the experiment process, the epoch was set to 60, and batch_size was set to 16 when using dataset 50/3,349, 100/3,299, and 200/3,199 to train the model. While batch_size was set to 4 when using dataset 25/3,374, because 25 labeled slices are too few, it was better to set batch_size to 4. The training strategy of SemiGAN was to train the generator 10 times before training the discriminator once to ensure GAN training stability using different training data settings.

(C) Co-GAN

The training dataset contains labeled and unlabeled data for semi-supervised learning based on Co-GAN. The training data settings were 25/3,374, 50/3,349, 100/3,299, and 200/3,199, notated as labeled slices/unlabeled slices. In this paper, the labeled and unlabeled datasets were fed into Generator 1, while the rotated labeled and rotated unlabeled datasets were fed into Generator 2. Co-GAN aimed to build two generators that can provide segmentation information for each other and used discriminators to distinguish between the predicted map and the ground truth. Through mutual guidance of two generators and adversarial training between generators and discriminators, Co-GAN was expected to achieve higher segmentation accuracy than F-Generator and Semi-GAN. The initialized two generators in Co-GAN were *Seg1only* and *Seg2only*, also used in the F-Generator experiment. Nonetheless, the parameter settings of generators and discriminators were the same as Semi-GAN.

### 3.3 Performance Evaluation

In this section, three evaluation metrics used in this paper are described. They includeed Dice, Hausdorff distance (HD), and relative volume error (RVE).

The Dice coefficient was used to evaluate the similarity between two samples. In medical image segmentation, the similarity or overlap between prediction result and ground truth can be calculated by the Dice function. A Dice value closer to 1 indicates a good segmentation result, while value closer to 0 shows a bad segmentation result. Given two sets A and B, the Dice index between them was defined as Eq. (14) [1],

$$Dice(A,B) = 2\frac{|A \cap B|}{|A|+|B|} \tag{14}$$

where $A$ and $B$ represent ground truth and predicted map, respectively.

Hausdorff distance describes the similarity between two sets of points. It also defines the distance between ground truth boundaries and predicted results. Furthermore, it was sensitive to the boundary of segmentation. Hausdorff distance is defined as Eq. (15) [1],

$$H = \max(\max_{i \in pr}(\min_{j \in gt}(d(i, j))), \max_{i \in gt}(\min_{j \in pr}(d(i, j)))) \tag{15}$$

where $i$ and $j$ are points belonging to ground truth and predicted map, respectively. Meanwhile, $d$ represents the distance between $i$ and $j$.

RVE describes the ratio of the absolute error of a measurement to the measurement being taken. In segmentation, it can be seen as an accuracy measurement. The lower the RVE is, the higher the segmentation accuracy is [35].

$$RVE(R_a, R_b) = \frac{abs(|R_a|-|R_b|)}{|R_b|} \tag{16}$$

where $R_a$ and $R_b$ represent ground truth and predicted map, respectively.

### 3.4 Discussion

The experiment results are shown in Tables 2-5. Table 2 shows the mean Dice value for F-Generator, Semi-GAN and Co-GAN learning models. As stated earlier, the segmentation results from two initialized generators are called Seg1only and Seg2only in F-Generator. Then, in Co-GAN, the segmentation results generated from Generator 1 is called Seg1CoGAN, and segmentation results generated from Generator 2 is called Seg2CoGAN. It should be noted that Semi-GAN only used Seg1only to be the initialized generator. Semi-GAN is used to justify the effect of using unlabeled data and is also used to explain the effect of mutual guidance between two generators, which is the co-training in Co-GAN.

Dice is a commonly used segmentation evaluation metric. The segmentation performance indicates better when the overlap of the predicted map and ground truth is closer to 1 and is relatively intuitive to evaluate segmentation results. Table 2 shows the Dice values using four data settings, and the value shown in parenthesis are the variation of Dice values distribution. Table 2 demonstrates that the utilization of unlabeled data during training improved image segmentation performance. On average, the Co-GAN model has the highest Dice value at 0.74025 (*Seg2CoGAN*). Meanwhile, Semi-GAN and F-Generator (*Seg1only*) scored Dice values of 0.72925 and 0.7255, respectively. When compared at the data settings level, our proposed Co-GAN achieved the highest Dice values of 0.688 (*Seg1CoGAN*) using 25/3,374 labeled/unlabeled data, 0.735 (*Seg2CoGA*N) at 50/3,349 labeled/unlabeled data, 0.774 (*Seg2CoGAN*) at 100/3,299 and 0.783 (*Seg1CoGAN*) at 200/3,199 labeled/unlabeled data. At the same time, Co-GAN achieved significantly higher Dice values compared to F-Generator and Semi-GAN under 25/3,374 labeled/unlabeled data and 50/3,349 labeled/unlabeled data. In contrast, Co-GAN improved slightly under 100/3,299 and 200/3,199 labeled/unlabeled data. It signifies that Co-GAN is more effective when using less labeled data.

Furthermore, the variation of Co-GAN scored the lowest value compared to Semi-GAN and F-Generator, which means that the Dice values distribution of Co-GAN has the average smallest degree of dispersion (0.0285) than F-Generator (0.032) and Semi-GAN (0.030). At each data settings level, our proposed Co-GAN model scored the lowest distribution value of 0.035 (*Seg1CoGAN*) using 25/3,374 labeled/unlabeled data, 0.028 (*Seg2CoGAN*) at 50/3,349 labeled/unlabeled data, 0.021 (*Seg2CoGAN*) at 100/3,299 and 0.025 (*Seg1CoGAN*) at 200/3,199 labeled/unlabeled data.

Overall, the semi-supervised models that are Semi-GAN and Co-GAN marginally outperformed the fully-supervised model F-Generator in segmenting the medical images. This shows that the unlabeled data can provide more information to the semi-supervised learning models to improve image segmentation.

Upon closer inspection, our proposed Co-GAN showed promising results in leveraging unlabeled data. The highest Dice value of Co-GAN at 0.783 by Seg1CoGAN was achieved using 200 labeled data and 3,199 unlabeled data. This is only slightly higher than the F-Generator model using 200 labeled data. When the Co-GAN model used 100 labeled data and 3,299 unlabeled data, it managed to get a Dice value of 0.774, which is marginally higher than the F-Generator model of 0.771 using 100 labeled data. Similarly, F-generator using 50 labeled data, Co-GAN using 25 labeled data, and 3,374 unlabeled data have comparable Dice values. Thus, the proposed Co-GAN can achieve similar results using less labeled data with the information of unlabeled data.

**Table 2. Mean dice values of F-Generator, Semi-GAN and Co-GAN.**

| Dataset settings (labeled) | F-Generator | | Dataset settings (labeled/ unlabeled) | Semi-GAN | Co-GAN | |
|---|---|---|---|---|---|---|
| | *Seg1only* | *Seg2only* | | *Seg1semiGAN* | *Seg1CoGAN* | *Seg2CoGAN* |
| 25 | 0.669<br>(0.040) | 0.329<br>(0.052) | 25/3,374 | 0.666<br>(0.039) | **0.688**<br>**(0.035)** | 0.677<br>(0.038) |
| 50 | 0.691<br>(0.037) | 0.727<br>(0.030) | 50/3,349 | 0.709<br>(0.035) | 0.717<br>(0.032) | **0.735**<br>**(0.028)** |
| 100 | 0.760<br>(0.025) | 0.771<br>(0.024) | 100/3,299 | 0.766<br>(0.022) | 0.771<br>**(0.021)** | **0.774**<br>(0.022) |
| 200 | 0.782<br>(0.026) | 0.771<br>(0.028) | 200/3,199 | 0.776<br>(0.026) | **0.783**<br>**(0.025)** | 0.775<br>(0.026) |
| Average | 0.7255<br>(0.032) | 0.6495<br>(0.0335) | | 0.72925<br>(0.0305) | 0.73975<br>**(0.0285)** | **0.74025**<br>(0.0285) |

To further describe the Dice values distribution in testing data, the number (Dice > 0.8) of the three models in Table 3 gives a statistical analysis for Dice values that are larger than 0.8. Meanwhile, the maximum Dice value of the three models in Table 4 can be used to show whether the experiments can improve the Dice value. Tables 3 and 4 show that Co-GAN improved the number of segmented data with high similarity Dice values. For example, Co-GAN segmented 112 and 120 testing data with Dice value larger than 0.8 successfully under dataset setting 50/3,349. While F-Generator just achieved 87 and 117 testing data segmentation using the same dataset setting in Table 3.

**Table 3. The number of (Dice > 0.8) of F-Generator, Semi-GAN and Co-GAN.**

| Dataset setting (labeled) | F-Generator | | Dataset setting (labeled /unlabeled) | Semi-GAN | Co-GAN | |
|---|---|---|---|---|---|---|
| | *Seg1only* | Seg2only | | *Seg1semiGAN* | *Seg1CoGAN* | *Seg2coGAN* |
| 25 | 85 | 0 | 25/3,374 | 82 | 90 | 75 |
| 50 | **87** | **117** | 50/3,349 | 112 | **112** | **120** |
| 100 | 136 | 146 | 100/3,299 | 138 | 139 | 148 |
| 200 | 168 | 164 | 200/3,199 | 165 | 166 | 167 |

**Table 4. Maximum Dice values of F-Generator, Semi-GAN and Co-GAN.**

| Dataset setting (labeled) | F-Generator | | Dataset setting (labeled/ unlabeled) | Semi-GAN | Co-GAN | |
|---|---|---|---|---|---|---|
| | *Seg1only* | *Seg2only* | | *Seg1semiGAN* | *Seg1CoGAN* | *Seg2CoGAN* |
| 25 | **0.918** | **0.744** | 25/3,374 | 0.915 | **0.922** | **0.925** |
| 50 | 0.933 | 0.934 | 50/3,349 | 0.929 | 0.929 | 0.936 |
| 100 | 0.947 | 0.928 | 100/3,299 | 0.949 | 0.948 | 0.938 |
| 200 | 0.929 | 0.954 | 200/3,199 | 0.928 | 0.933 | 0.955 |

In Table 4, the maximum Dice values in Co-GAN are 0.922 and 0.925 using dataset setting 25/3,374, while F-Generator scored maximum Dice values of 0.918 and 0.744 using the same dataset setting. This means Co-GAN is effective when more information is gathered from unlabeled data. Table 5 illustrates the relative volume error (RVE) values of the three models, and the value in the parenthesis is the variation of RVE values distribution. The Co-GAN model scored the lowest average RVE of 0.3105, while F-Generator has the highest RVE of 0.41255. This indicates that Co-GAN has a higher accuracy than F-Generator in performing image segmentation. The Co-GAN model outperformed F-Generator and Semi-GAN at the data settings level using 25/3,374 and 50/3,349 labeled/unlabeled data. However, the RVE of the F-Generator was smaller than Semi-GAN and Co-GAN at data settings of 100/3,299. Finally, at data settings 200/3,199, Semi-GAN scored the lowest RVE at 0.308. At the same time, Co-GAN can achieve significantly lower RVE values compared to F-Generator and Semi-GAN under 25/3,374 labeled/unlabeled data and 50/3,349 labeled/unlabeled data. It means Co-GAN is more effective when using less labeled data. Even though the results of RVE are a bit inconsistent, semi-supervised models of Semi-GAN and co-GAN showed that the use of unlabeled data during the training could improve image segmentation. Further investigation needs to be done to improve the semi-supervised models further.

Based on the results shown in Tables 2-5, the segmentation performance of Co-GAN model is better than F-Generator by comparing the different methods under one data setting. The variations of Dice values are reduced when using unlabeled data. The experiment results show that unlabeled data can improve segmentation accuracy. While in the experiments, the time cost of the proposed method is affected by the hardware, the size of model and input data. The experiments were conducted on Kaggle platform, thus the training time is mainly affected by model size and input data in this paper. The Co-GAN has two generators and needs two data views for training, which will cost more training time compared to semi-GAN and F-generator.

**Table 5. RVE values of F-Generator, Semi-GAN and Co-GAN.**

| Dataset setting (labeled) | F-Generator | | Dataset settings (labeled/ unlabeled) | Semi-GAN | Co-GAN | |
|---|---|---|---|---|---|---|
| | *Seg1only* | *Seg2only* | | *Seg1semiGAN* | *Seg1coGAN* | *Seg2CoGAN* |
| 25 | 0.340<br>(0.188) | 0.716<br>(0.007) | 25/3,374 | 0.342<br>(0.196) | 0.336<br>(0.211) | **0.319**<br>(0.150) |
| 50 | 0.443<br>(0.824) | 0.345<br>(0.426) | 50/3,349 | 0.322<br>(0.312) | 0.321<br>(0.420) | **0.312**<br>(0.505) |
| 100 | 0.328<br>(0.565) | **0.308**<br>(0.488) | 100/3,299 | 0.325<br>(0.603) | 0.321<br>(0.597) | 0.319<br>(0.564) |
| 200 | 0.279<br>(0.376) | 0.280<br>(0.375) | 200/3,199 | **0.276**<br>(0.360) | 0.279<br>(0.360) | 0.292<br>(0.416) |
| Average | 0.3475<br>(0.48825) | **0.41255**<br>**(0.324)** | | 0.31625<br>(0.36775) | 0.31425<br>(0.397) | **0.3105**<br>(0.40875) |

Fig. 9 represents the segmentation results of one slice. The segmentation results of different experiments under four training data settings are presented for this slice. It can be seen that more labeled data produced better segmentation results. When comparing the results of different experiments under the same dataset setting, the segmentation results must be compared by using evaluation metrics. The main reason might be due to the generators in Co-GAN that adopted the initialized generators in F-Generator, which has been optimal. Thus, generators in Co-GAN must be modified to get clearer improvements in the future.

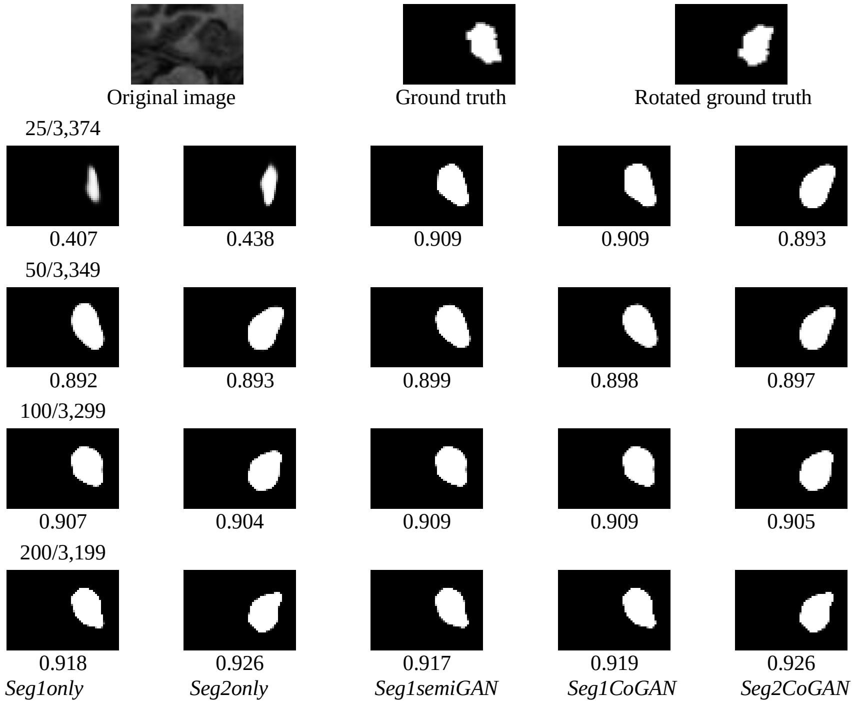

Fig. 9. The segmentation results of different methods.

## 4. CONCLUSIONS

This paper proposes semi-supervised learning based on Co-GAN to implement medical image (anterior hippocampus) segmentation. Based on the experiment results analysis, semi-supervised learning based on Co-GAN can achieve better segmentation performance by using co-training between two generators and an adversarial training strategy between the generators and discriminator. In the discriminator, not only the target region information is used, but the background information also provides help in training. According to the results of the comparative experiments, semi-supervised learning based on Co-GAN can utilize unlabeled data to enhance segmentation performance. Moreover, Co-GAN is more effective when using less labeled data. However, according to the present experiments that we done, it is necessary to retrain and tune the parameters to adapt different datasets. And the initialized generators must be modified in the future to achieve higher improvements. In the next study, we aim to study and optimize the initialization of generators and try to segment more different medical images.

## REFERENCES

1. X. Liu, L. Song, S. Liu, and Y. Zhang, "A review of deep-learning-based medical image segmentation methods," *Sustainability*, Vol. 13, 2021, p. 1224.
2. Z. Cheng, A. Qu, and X. He, "Contour-aware semantic segmentation network with spatial attention mechanism for medical image," *The Visual Computer*, Vol. 38, 2021, pp. 749-762.
3. D. Wang, G. Hu, and C. Lyu, "Multi-path connected network for medical image segmentation," *Journal of Visual Communication and Image Representation*, Vol. 71, 2020, p. 102852.
4. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, Vol. 42, 2017, pp. 60-88.
5. M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical Image Analysis*, Vol. 35, 2017, pp. 18-31.
6. X. Qiu, "A new multilevel feature fusion network for medical image segmentation," *Sensing and Imaging*, Vol. 22, 2021, pp. 1-20.
7. Y. Sun, J. Tang, W. Lei, and D. He, "3D segmentation of pulmonary nodules based on multi-view and semi-supervised," *IEEE Access*, Vol. 8, 2020, pp. 26457-26467.
8. Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen, "Deep adversarial networks for biomedical image segmentation utilizing unannotated images," in *Proceedings of the 20th International Conference on Medical Image Computing and Computer Assisted Intervention*, 2017, pp. 408-416.
9. D. Nie, Y. Gao, L. Wang, and D. Shen, "ASDNet: attention based semi-supervised deep networks for medical image segmentation," in *Proceedings of the 21st International Conference on Medical Image Computing and Computer Assisted Intervention*, 2018, pp. 370-378.

10. W. Bai, O. Oktay, M. Sinclair, H. Suzuki, M. Rajchl, G. Tarroni, B. Glocker, A. King, P. M. Matthews, and D. Rueckert, "Semi-supervised learning for network-based cardiac MR image segmentation," in *Proceedings of the 20th International Conference on Medical Image Computing and Computer Assisted Intervention*, 2017, pp. 253-260.
11. Z. Feng, D. Nie, L. Wang, and D. Shen, "Semi-supervised learning for pelvic MR image segmentation based on multi-task residual fully convolutional networks," in *Proceedings of IEEE 15th International Symposium on Biomedical Imaging*, 2018, pp. 885-888.
12. J. Xing, Z. Li, B. Wang, Y. Qi, B. Yu, F. G. Zanjani, and A. Zheng, "Lesion segmentation in ultrasound using semi-pixel-wise cycle generative adversarial nets," *IEEE/ ACM Transactions on Computational Biology and Bioinformatics*, Vol. 18, 2020, pp. 2555-2565.
13. A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 1998, pp. 92-100.
14. Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth, "3D semi-supervised learning with uncertainty-aware multi-view co-training," in *Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3646-3655.
15. J. Peng, G. Estrada, M. Pedersoli, and C. Desrosiers, "Deep co-training for semi-supervised image segmentation," *Pattern Recognition*, Vol. 107, 2020, p. 107269.
16. C. Tseng, T. Huang, and T. Liu, "Data labeling with novel decision module of tri-training," in *Proceedings of the 2nd International Conference on Computer Communication and the Internet*, 2020, pp. 82-87.
17. Z. Li, L. Lin, C. Zhang, H. Ma, and W. Zhao, "Automatic image annotation based on co-training," in *Proceedings of International Joint Conference on Neural Networks*, 2019, pp. 1-8.
18. S. Min and X. Chen, "A robust deep attention network to noisy labels in semi-supervised biomedical segmentation," *arXiv Preprint*, 2022, arxiv:2204.02961.
19. R. Azad, N. Khosravi, and D. Merhof. "SMU-Net: Style matching U-Net for brain tumor segmentation with missing modalities," *arXiv Preprint,* 2022, arXiv: 2204. 02961.
20. Y. Zhou, Y. Wang, P. Tang, S. Bai, W. Shen, E. Fishman, and A. Yuille, "Semi-supervised 3D abdominal multi-organ segmentation via deep multi-planar co-training," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 121-140.
21. T. S. Abdelgayed, W. G. Morsi, and T. S. Sidhu, "Fault detection and classification based on co-training of semi supervised machine learning," *IEEE Transactions on Industrial Electronics*, Vol. 65, 2018, pp. 1595-1605.
22. L. Du, Y. Wang, and W. Xie, "A semi-supervised method for SAR target discrimination based on co-training," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 9482-9485.
23. B. Zhou, Y. Wang, W. Liu, and B. Liu, "Identification of working condition from sucker-rod pumping wells based on multi-view co-training and hessian regularization of SVM," in *Proceedings of the 14th IEEE International Conference on Signal Processing*, 2018, pp. 969-973.

24. X. Duan, N. Thomsen, Z. Tan, B. Lindberg, and S. Jensen, "Weighted score based fast converging CO-training with application to audio-visual person identification," in *Proceedings of IEEE 29th International Conference on Tools with Artificial Intelligence*, 2017, pp. 610-617.
25. X. Ning, X. Wang, S. Xu, S. Xu, W. Cai, L. Zhang, L. Yu, and W. Li, "A review of research on co-training," *Concurrency and Computation: Practice and Experience*, Wiley, 2021.
26. I. Goodfellow, J. Pouget-Abadie, M. Mirza, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, Vol. 27, 2014, pp. 2672-2680.
27. S. Xun, D. Li, H. Zhu, M. Chen, J. Wang, M. Chen, B. Wu, H. Zhang, X. Chai, Z. Jiang, Y. Zhang, and P. Huang, "Generative adversarial networks in medical image segmentation: A review," *Computers in Biology and Medicine*, Vol. 140, 2020, No. 205063.
28. D. Nie and D. Shen, "Adversarial confidence learning for medical image segmentation and synthesis," *International Journal of Computer Vision*, Vol. 128, 2020, pp. 2494-2513.
29. G. Li, N. Jamil, and R. Hamzah, "Leveraging unlabeled data using semi-supervised generative adversarial network for medical image segmentation," in *Proceedings of International Conference on Computer and Drone Applications*, 2022, pp. 67-72
30. A. Lahiri, K. Ayush, P. K. Biswas, and P. Mitra, "Generative adversarial learning for reducing manual annotation in semantic segmentation on large scale miscroscopy images: Automated vessel segmentation in retinal fundus image as test case," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 42-48.
31. D. Yang, D. Xu, S. K. Zhou, B. Georgescu, M. Chen, S. Grbic, D. Metaxas, and D. Comaniciu, "Automatic liver segmentation using an adversarial image-to-image network," in *Proceedings of the 20th International Conference on Medical Image Computing and Computer Assisted Intervention*, 2017, pp. 507-515.
32. D. Jin, Z. Xu, Y. Tang, A. P. Harrison, and D. J. Mollura, "CT-realistic lung nodule simulation from 3D conditional generative adversarial networks for robust lung segmentation," in *Proceedings of the 21st International Conference on Medical Image Computing and Computer Assisted Intervention*, 2018, pp. 732-740.
33. Y. Zhou, X. He, L. Huang, L. Liu, F. Zhu, S. Cui, and L. Shao, "Collaborative learning of semi-supervised segmentation and classification for medical images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2079-2088.
34. L. Han, Y. Huang, H. Dou, S. Wang, S. Ahamad, H. Luo, Q. Liu, J. Fan, and J. Zhang, "Semi-supervised segmentation of lesion from breast ultrasound images with attentional generative adversarial network," *Computer Methods and Programs in Biomedicine*, Vol.189, 2020, No. 105275.
35. G. Gerig, M. Jomier, and M. Chakos, "Valmet: a new validation tool for assessing and improving 3D object segmentation," in *Proceedings of the 4th International Conference Medical Image Computing and Computer-Assisted Intervention*, 2001, pp. 516-523.

**Guo-Qin Li (李国琴)** was born in 1992. She received the MS degree in Signal and Information Processing from Northwestern Polytechnical University. She is currently a Lecturer in Taiyuan Institute of Technology. She is studying for her Ph.D. degree in College of Computing, Informatics and Media in Universiti Teknologi MARA (UiTM), Shah Alam, Malaysia. Her current research interests include medical image processing based on deep learning. Her email is liguoqin@studyedus.cn.

**Nursuriati Jamil** is a Professor in College of Computing, Informatics and Media in Universiti Teknologi MARA (UiTM), Shah Alam, Malaysia. Her research path began with multimedia information retrieval moving into image processing and speech analysis and is now focusing on applying machine learning algorithms to solve real-life problems. She founded Digital Image, Audio and Speech Technology research group in her university and is now heading the National Autism Resource Centre. Over the last five years, she has secured over RM1 million research grants and has attained h-index of 15 and 1,067 citations. Nursuriati is a senior member of IEEE and a certified professional of Malaysia Board of Technologist.

**Raseeda Hamzah** has been a Lecturer in Universiti Teknologi MARA since 12 August 2016. She is currently a Senior Lecturer at the College of Computing, Informatics and Media in Universiti Teknologi MARA (UiTM), Melaka Branch, Malaysia. Secured her Ph.D. in Information Technology and Quantitative Sciences, Universiti Teknologi Mara, Cawangan Shah Alam. Her area of expertise is digital signal processing specifically in speech analysis and image processing. She is also actively doing research in urban farming and IoT. Her email is raseeda@uitm.edu.my.