Data Augmentation of Thyroid Ultrasound Images Using Generative Adversarial Network

Junzhao Liang and Junying Chen*
School of Software Engineering, South China University of Technology
Key Laboratory of Big Data and Intelligent Robot (South China University of Technology), Ministry of Education
Guangzhou, China
Email: 202021045572@mail.scut.edu.cn, jychense@scut.edu.cn

2021 IEEE International Ultrasonics Symposium (IUS) | 978-1-6654-0355-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/IUS52206.2021.9593403

***Abstract*—Ultrasound (US) has been investigated as a common method of computer aided diagnosis because of its low-cost, harmless and real-time scanning. Also the rapid development of deep learning segmentation and classification models alleviates the influence of low signal-to-noise ratio and artifacts of ultrasonic imaging. However, due to the privacy issues of medical data, it is not easy to acquire sufficient data for deep learning model training. In recent years, generative adversarial networks (GANs) are widely used in data augmentation. However, GANs suffer from the problem of mode collapse in the training process then generate images with a limited variety. On the other hand, variational auto-encoder (VAE) is free from mode collapse but it generates blurred images. In this work, we study an auto-encoding generative adversarial network combining the advantages of GAN and VAE to generate realistic images for medical thyroid ultrasound image augmentation. Experiment results show that the generated images can simulate realistic ultrasound features and thyroid tissues for augmentation and help training a U-Net model to get better segmentation results.**

***Index Terms*—Generative adversarial network, data augmentation, ultrasound, thyroid**

I. Introduction

US imaging has the advantages such as low cost and harmlessness, as compared to other kinds of medical imaging, e.g., magnetic resonance imaging, computed tomography, and positron emission tomography. However, the US imaging quality is limited due to the device development. Futhermore, because of the privacy issues of medical data which contains the information of patients, it is not easy to acquire a large amount of US images.

With the great success of convolutional neural networks (CNNs), recent research studies ultilize CNN-based models in image processing. These models can perform fast, real-time and accurate segmentation [1] and classification [2]. For example, a well-trained model can locate thyroid area and mark out its boundary on a US image in a short time while handcrafted segmentation is time-consuming. So that fast segmentation can accerlerate the diagnosis process. However, training a robust deep learning model requires a sufficient and variant dataset, otherwise the trained model would overfit on the training data and perform bad results in application [3].

This work was supported in part by the National Natural Science Foundation of China under Grant 61802130, and in part by the Guangdong Natural Science Foundation under Grant 2019A1515012152 and Grant 2021A1515012651.

*Corresponding author.

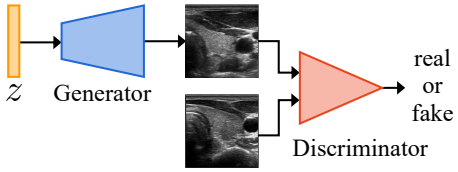


Fig. 1: The architecture of GAN.

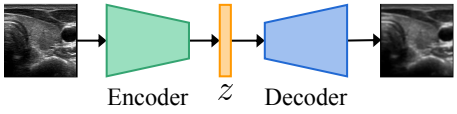


Fig. 2: The architecture of VAE.

Conventional data augmentation methods (e.g., flip, rotate, resize, elastic distort) based on affine transformation are widely ultilized to enhance the robustness of deep learning models. But such augmentation methods play a limited role in increasing the variety of data and narrowing the gap between real and training data distribution, because the transformed images are highly related to the training data. Recently, GANs [4] are widely ultilized in vector-to-image generation tasks [3], [5], [6] and image-to-image translation tasks [7], [8] due to their powerful generation capacity. GANs can learn features from all training data and then produce new realistic images by combining these features. As a consequence, the generated images by GANs can increase the variety of the training data.

A GAN model consists of a generator and a discriminator as shown in Fig. 1. These two parts play an adversarial learning during model training. After that, the generator can simulate realistic images from prior distribution in the inference process. The special training strategy and loss functions of the vanilla GAN create unstable training gradients and cause the GAN model to converge in a local optimum, i.e. mode collapse [9]. When mode collapse occurs, the GAN model concentrates on a few of tarining data and learns limited features such that the generator keeps generating realistic images but with high similarity. In extreme circumstances, all the generated

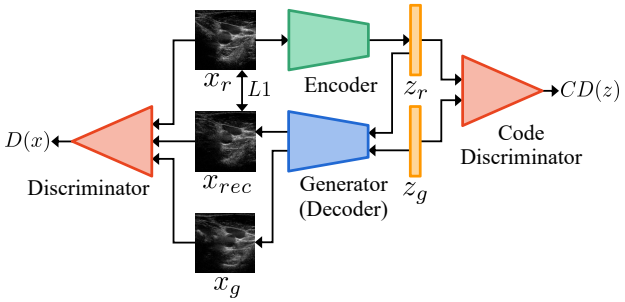


Fig. 3: The architecture of our mdoel.

images are identical to each other. And it blocks the training process because the discriminator cannot discriminate the generated images from real images. The generator is well-trained in the discriminator's view and thus the discriminator provides useless gradients. VAE [10] is a substitution of GAN which embeds images into a specific prior distribution by reconstructing them with an encoder and a decoder as shown in Fig. 2. It is free from mode collapse but it generates images with blurriness by using L1 loss in the reconstruction process.

α-GAN [11] adopts the idea of VAE and introduces an encoder and a code discriminator into the framework of GAN to embed real images into the prior distribution which is benefit for stablizing the training. The code discriminator does similar adversarial learning with the encoder like what the discriminator does with the generator, and treats the embeded vectors as fake and the random sampled vectors as real. And the variational inference is replaced by the code discrimination. α-WGAN-GP [3] ultilizes the architecture of α-GAN and the loss functions of Wasserstein GAN with gradient penalty (WGAN-GP) [12] to address the mode collapse and blurriness problems. However, α-WGAN-GP discards the backbone of WGAN-GP and implements batch normalization (BN) [13] in the discriminator which is against the set up of WGAN-GP [12]. Moreover, the original design of α-WGAN-GP utilizes 3D convolution layers which does not directly fit on 2D thyroid US images.

In this paper, we propose an improved α-WGAN-GP adopting the backbone of WGAN-GP and implementing instance normalization (IN) in the discrimiator to ensure the training stability and get rid of the blurred images. The convolution layers of our model are modified from 3D to 2D to match the inputs. Our model is trained to generate realistic images from random vectors on a self-collected thyroid US dataset. Then we use different numbers of generated realistic images and their corresponding handcrafted labels to augment the original dataset and use the augmented datasets for U-Net training. Finally, we conduct qualitative and quantitative evaluations including the comparasion with baseline models and improvement of U-Net with different augmented datasets to verify the effectiveness of our model.

II. Methods

A. Architecture

The proposed model consists of four parts as shown in Fig. 3, including a discriminator D, a generator (i.e. decoder) G, an encoder E and a code discriminator CD. In the training process, E embeds the real images into a posterior distribution, a low-dimensional latent space. Then the CD discriminates the embeded vectors z_r from the random vectors z_g sampled from a specific prior distribution $P(z)$, e.g., Gaussian distribution. z_r and z_g are treated as fake and real by CD respectively. When CD can no longer recognize z_r, we assume that the real images x_r are perfectly embeded into $P(z)$ and so that we can sample any vectors representing real images from $P(z)$ unlimitedly. G learns the mapping from z_g to x_r by reconstructing x_r with z_r and playing adversarial training with D. Reconstructed images x_{rec} and generated images x_g are treated as fake by D. The discrimination roles of D and CD are the same except the inputs are images and vectors respectively. In the inference process, only G is used to generate realistic images while E, CD and D are discarded.

There are seven convolution layers in G, D and E. All the convolution layers of G use 3×3 filters except the first transpose convolution layer uses 4×4 filter. All the convolution layers of D and E use 4×4 filters. BN and rectified linear unit (ReLU) are implemented in the $1\sim6$ layers of G. IN and Leaky ReLU are implemented in the $2\sim6$ and $1\sim6$ layers of D respectively, because using BN in D changes the penalized training objective of WGAN-GP [12]. E shares the same architecture as D but uses BN in $1\sim6$ layers instead of IN. CD is a multilayer perceptron consisting of 3 fully-connected layers. Leaky ReLU is implemented in the $1\sim2$ layers of CD.

B. Loss Functions

The loss functions of the original GAN are as follows:

$$\begin{aligned} L_D &= -\mathbb{E}[\log(D(x_r))] - \mathbb{E}[\log(1 - D(x_g))], \\ L_G &= \mathbb{E}[\log(1 - D(x_g))], \end{aligned} \tag{1}$$

where x_r and x_g represent real and fake images, respectively. x_g is generated by $G(z_g)$ where random vector z_g is sampled from a prior distribution. Using (1) in the proposed model unstablizes the training process and results in mode collapse. Hence, we ultilize the loss functions of WGAN-GP, which are:

$$\begin{aligned} L_D &= -\mathbb{E}[D(x_r)] + \mathbb{E}[D(x_g)] + \lambda L_{GP}, \\ L_G &= -\mathbb{E}[D(x_g)], \\ L_{GP} &= \mathbb{E}[(\|\nabla D(x_p)\|_2 - 1)^2], \end{aligned} \tag{2}$$

where λ is the empirically determined weight of L_{GP}, L_{GP} is the gradient penalty of discriminator, x_p is the weighted average of x_r and x_g. L_{GP} is ultilized to maintain the 1-lipschitz assumption to ensure the training stability together with the wasserstein distance losses.

E and CD, G and D can be treated as two pairs of GANs. Hence, the final loss functions of the proposed model are as follows:

$$\begin{aligned} L_D =& -2\mathbb{E}[D(x_r)] + \mathbb{E}[D(x_g)] \\ &+ \mathbb{E}[D(x_{rec})] + \lambda_1 L_{GP1}, \\ L_G =& -\mathbb{E}[D(x_g)] - \mathbb{E}[D(x_{rec})] + \lambda_2 \|x_r - x_g\|_{L1}, \\ L_{CD} =& -\mathbb{E}[CD(z_g)] + \mathbb{E}[CD(z_r)] + \lambda_1 L_{GP2}, \\ L_E =& -\mathbb{E}[CD(z_r)], \\ L_{GP1} =& \mathbb{E}[(\|\nabla D(x_p)\|_2 - 1)^2], \\ L_{GP2} =& \mathbb{E}[(\|\nabla CD(z_p)\|_2 - 1)^2], \end{aligned} \tag{3}$$

where λ_1 and λ_2 are the empirically determined weight of gradient panelty and L1 loss between real images x_r and their reconstructed images x_{rec}, respectively. $\lambda 1$ and $\lambda 2$ are both set to 10. z_r and z_g represent the embeded and sampled vectors, respectively. z_p is the weighted average of z_r and z_g. The dimensions of z_r and z_g are both set to 256. The negative slopes of Leaky ReLU are all set to 0.2. The parameters of CD, E, D and G are updated in order in a training iteration.

III. Experiments and Results

A. Datasets

Dataset A consists of 609 thyroid US images without the corresponding segmentation labels. We use the dataset A for GAN training. We right-aligned crop the original 875×1167 images as shown in Fig. 4a into 875×875 and then resize them into 256×256 as shown in Fig. 4b to reduce the GPU memory usage of model training.

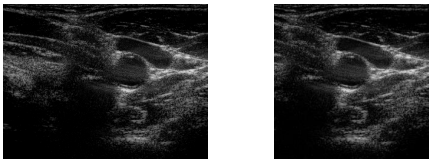
Dataset B consists of 190 thyroid US images and their corresponding segmentation labels as shown in Fig. 5a and Fig. 5b, respectively. We randomly select 128 images for training and the left 62 for testing.

Conventional data augmentation methods are not used on above datasets.

B. Implementation Details

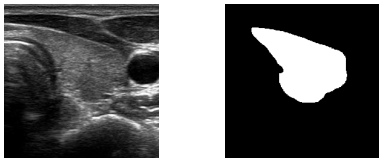
Our experiments are implemented with Pytorch library on an NVIDIA RTX 2080Ti 11GB GPU. All images are normalized to the range [-1,1] before being input into the model. The Adam optimizers with learning rate of 0.0002 and 0.01 are implemented in augmentation and segmentation tasks respectively while the learning rate lr of segmentation task reduces to $0.1\times lr$ every 20 epochs. The batch sizes of input images are 32 and 16 while the training epochs are 5000 and 60 for augmentation and segmentation tasks, respectively.

In generation task, we use Maximum Mean Discrepancy (MMD) score and 1-Nearest Neighbor (1-NN) score to investigate the distance between real and generated images distribution. MMD and 1-NN are calculated by averaging values of 10 test with 100 randomly selected images from each distribution. Moreover, we use Multi-Scale Structural Similarity (MS-SSIM) index to investigate the similarity among generated images. MS-SSIM is calculated by averaging 10 test among 32 randomly selected images. The MS-SSIM index of dataset A is 0.3129. In segmentation task, we use Mean Intersection over Union (mIoU) and Dice score to investigate the segmentation results.



(a) (b)

Fig. 4: A sample from dataset A. (a) original sample. (b) processed sample.



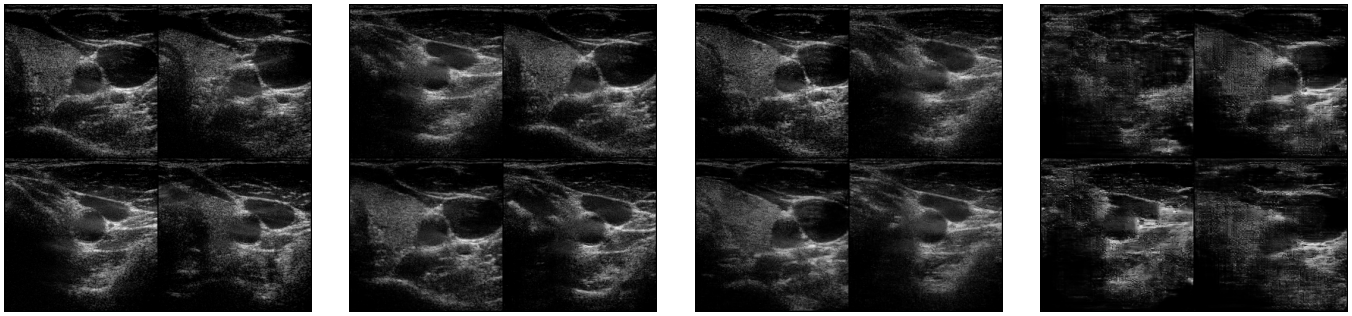
(a) (b)

Fig. 5: A sample pair of dataset B. (a) US thyroid image. (b) segmentation label.

C. Results

α-WGAN-GP [3] and WGAN-GP [12] are used as baseline models for comparasion of the generation task. We train our model and baseline models on dataset A to evaluate their generation capacity. The visualization of generation results of each models are shown in Fig. 6. TABLE I shows the quantitative results. Our model outperforms the other two baseline models with 9.09% and 31.86% improvement on MMD and 3.15% and 37.17% improvement on 1-NN scores respectively which indicates that the generated images distribution of our model well matches the real images distribution. Moreover, our model gets the second lowest MS-SSIM index which indicates that our generated images are rich in variety as shown in Fig. 6b. Although α-WGAN-GP gets the lowest MS-SSIM index, its 1-NN score is extremely high comparing to the other two models. It indicates that the generated images from α-WGAN-GP as shown in Fig. 6d are not close to the real images and obvious artifacts can be found in the images. The results above demonstrate that introducing the architecture of α-WGAN-GP into WGAN-GP improve the generation capacity and using the backbone of WGAN-GP in α-WGAN-GP with the loss functions of WGAN-GP guarantees the training stability. Thus, our model is able to learn the features of training images and then generates realistic images with diverse features.

For the segmentation task, we first train our model on dataset B with the same settings in genertaion task. Then we randomly generate 64 thyroid US images through the trained model and segment these 64 images by hand for labels. 0, 32, 64 of the generated images with their corresponding labels are used to augment the training set of dataset B respectively. These three different augmented training sets are used for training a U-Net segmentation network to evaluate the effect of augmentation. TABLE II shows the quantitative results which



(a) real images (b) Ours (c) WGAN-GP (d) α-WGAN-GP

Fig. 6: The real images and the generated images.

TABLE I: Quantitative results of generation task

Methods \ Metrics	MMD	1-NN	MS-SSIM
WGAN-GP	0.1199	0.6322	0.3713
α-WGAN-GP	0.1600	0.9746	**0.3000**
Ours	**0.1090**	**0.6123**	0.3445

TABLE II: Quantitative results of segmentation task

Datasets \ Metrics	mIoU	Dice
128	0.3881	0.5093
128+32	0.3885	0.5105
128+64	**0.3892**	**0.5111**

demonstrate that U-Net performs better segmentation results when more generated images are used in training because our generated images are realistic and diverse enough to increase the variety of the training images and help U-Net to learn a more robust representation. We attribute the little improvement of segmentation to the lack of labels of more generated images for time-consuming handcrafted segmentation.

IV. Conclusion

In this paper, we propose an improved α-WGAN-GP adopting the backbone of WGAN-GP for thyroid US image augmentation and then train a U-Net for segmentation with augmented datasets. The experiment results demonstrate that our model can be used for data augmentation when the quantity of avaliable US images are limited which is common in medical deep learning research. Futhermore, the generated images can be used for training a segmentation model for better results which improves the real-time US diagnosis. The future work may be generating pairs of images and labels to get rid of the handcrafted segmentation of generated images.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, 2015.

[2] H.-C. Shin, A. Ihsani, Z. Xu, S. Mandava, S. T. Sreenivas, C. Forster, J. Cha, *et al.*, "Gandalf: Generative adversarial networks with discriminator-adaptive loss fine-tuning for alzheimer's disease diagnosis from mri," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 688–697, 2020.

[3] G. Kwon, C. Han, and D.-s. Kim, "Generation of 3d brain mri using auto-encoding generative adversarial networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 118–126, 2019.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[5] C. Bermudez, A. J. Plassard, L. T. Davis, A. T. Newton, S. M. Resnick, and B. A. Landman, "Learning implicit brain mri manifolds with deep learning," in *Proc. SPIE 10574, Medical Imaging 2018: Image Processing*, p. 105741L, 2018.

[6] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, and H. Nakayama, "Gan-based synthetic brain mr image generation," in *IEEE International Symposium on Biomedical Imaging*, pp. 734–738, 2018.

[7] M. Engin, R. Lange, A. Nemes, S. Monajemi, M. Mohammadzadeh, C. K. Goh, T. M. Tu, B. Y. Tan, P. Paliwal, L. L. Yeo, *et al.*, "Agan: An anatomy corrector conditional generative adversarial network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 708–717, 2020.

[8] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *International Workshop on Simulation and Synthesis in Medical Imaging*, pp. 1–11, 2018.

[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, pp. 214–223, 2017.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, pp. 1–14, 2014.

[11] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, "Variational approaches for auto-encoding generative adversarial networks," *arXiv preprint arXiv:1706.04987*, 2017.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448–456, 2015.