Metal Artifact Reduction in CT Images using CycleGAN

Yu Zhou **Yage Hao** **Shuli Zhang** **Cai Tianzhang**

Abstract

In this project, we intend to reduce the metal artifact using cycleGAN. Specifically, the generators are splitted into several blocks, which can be assembled to achieve different functions. We also improve the generator in the cycleGAN by introducing a **artifact fusing** protocal, which will facilitate the generator to learn the features. CycleGAN doesn't require the paired data, which is indispensable in the traditional supervised MAR method. Meanwhile, it has better performance in terms of SSIM.

1 Introduction

Metal artefact is one of the commonly encountered problems in computed tomography (CT). It arises when a patient carries metallic implants, which will introduce severe streaking and shading artifacts in the reconstructed CT images. In this project, we intend to reduce the metal artifact using cycleGAN.

2 Related Work

A number of deep learning methods to address the metal artifact reduction problem has been proposed in recent years. cGANMAR proposed by Wang et.al. utilizes the structure of pix2pix to transfer the CT image from artifact-affect domain to artifact-free domain using a supervised way. (1). CNNMAR proposed by zhang et.al (2) combines the traditional algorithm as well as the convolutional neural network to reduce the streaked artifact around the metal. Some unsupervised methods have also been introduced. ADN (3) doesn't require paired data and can remove the artifact and add it on the artifact-free image at the same time.U-DuDoNet (4) extends the original DuDoNet method (5) and process the data both on sinogram domain and image domain.

3 Data

We use our own data acquired from the GuLou Hospital,Jiangsu,People Republic of China. The whole dataset contains 500 CT scan pictures(590×590 pixels). We take the first 400 samples as training set and remaining as test set.
We have done some proprosess work. First, we downsampled the image size to 256×256. Then some augmentation has been done, like reshuffle,centered cropping and randomly flipped.

4 Methods

4.1 Vanilla CycleGAN

The normal cycleGAN is shown as follows:

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

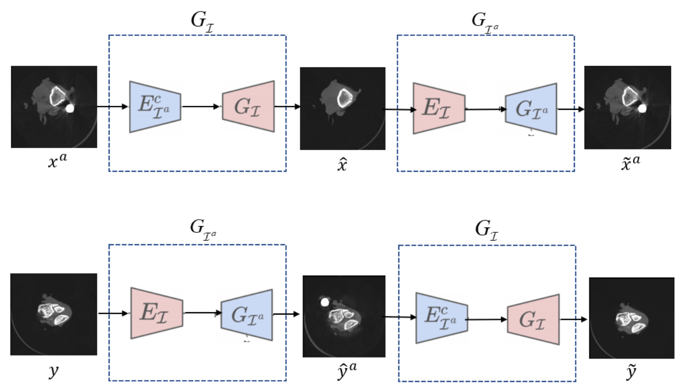

Figure 1: vanilla CycleGAN

In general, the network has two generators:$G_{\mathcal{I}}$, $G_{\mathcal{I}^a}$ and two discriminators:$D_{\mathcal{I}}$,$D_{\mathcal{I}^a}$. (6) $G_{\mathcal{I}}$ is designed to transfer image from domain $\mathcal{I}^a$ to $\mathcal{I}$ while $G_{\mathcal{I}^a}$ is designed to transfer the image from domain $\mathcal{I}$ to domain $\mathcal{I}^a$.($\mathcal{I}^a$ denotes the domain with artifact, and $\mathcal{I}$ denotes the domaini without artifact)
There are totally 2 kinds of encoders and 2 kinds of decoders:

- $E^c_{\mathcal{I}^a}$ Encoder that entangle the image from artifact image domain to feature latent
- $G_{\mathcal{I}}$ Decoder that distangle the image from feature latent to artifact-free domain.
- $E_{\mathcal{I}}$ Encoder that entangle the image from artifact-free image domain to feature latent.
- $G_{\mathcal{I}^a}$ Decoder that tranfer the feature latent to the image with artifact.

The encoder is formed by downsampling block and decoder is made up with upsampling block. The concrete configuration is presented in appendix A.
The simplified schematic with discriminator is shown below:

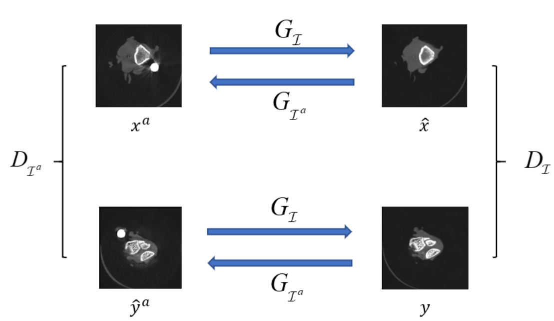

Figure 2: Schematic with Discriminator

The role of discriminator is just to distinguish the image sampled in dataset from the image formed by generator. The architecture is a series of convolutional block. It is copied from this repo.

4.2 CycleGAN with FPN structure

The image below shows the overal network architecture:

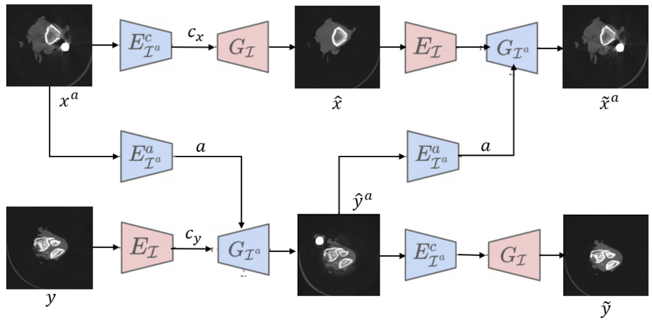

Figure 3: Structure of the network

Compared to the vanilla cycleGAN, we add an encoder $E^a_{\mathcal{I}^a}$ to get artifact features a from x^a and let generator $G_{\mathcal{I}^a}$ makes full use of a:

- $E^a_{\mathcal{I}^a}$ Encoder that entangle the image from artifact image domain to artifact latent, which can extract the artifact information.

And $G_{\mathcal{I}^a}$ now receives 2 inputs: artifact a and image features c_y:

- $G_{\mathcal{I}^a}$ Decoder that fuse the feature latent and artifact latent together to generate the image with artifact.

Actually, the combination of $E^a_{\mathcal{I}^a}$ and $G_{\mathcal{I}^a}$ is called Featured Pyramid Network, which is firstly proposed by Lin et.al in 2017 (7) and applied by ADN network (3):

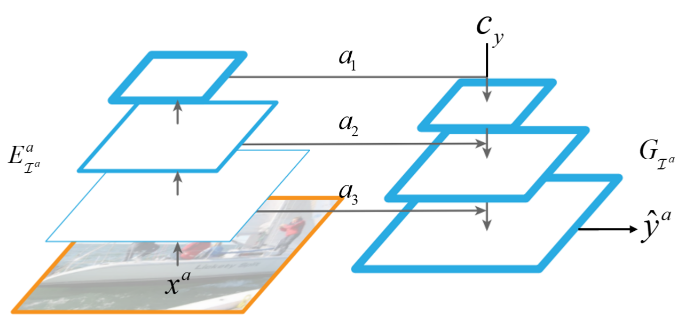

Figure 4: Feature pyramid network

At the encoder step,x^a is downsampled to different scales and generate multidimensional artifact features. Then we concatenate them to the features map of c_y along the channel direction, followed by a 1×1 convolution in the upsampling period.

4.3 Loss analysis

When it comes to loss function, we mainly have two parts: adversarial gan loss, cyclical consistency loss.
The first part is the vanilla gan loss:

$$\mathcal{L}^{\mathcal{I}}_{adv} = \mathbb{E}_{\mathcal{I}} [\log D_{\mathcal{I}}(y)] + \mathbb{E}_{\mathcal{I}^a} [1 - \log D_{\mathcal{I}}(\hat{x})] \tag{1}$$

$$\mathcal{L}^{\mathcal{I}^a}_{\text{adv}} = \mathbb{E}_{\mathcal{I}^a} [\log D_{\mathcal{I}^a} (x^a)] + \mathbb{E}_{\mathcal{I},\mathcal{I}^a} [1 - \log D_{\mathcal{I}^a} (\hat{y}^a)] \tag{2}$$

$$\mathcal{L}_{\text{adv}} = \mathcal{L}^{\mathcal{I}}_{\text{adv}} + \mathcal{L}^{\mathcal{I}^a}_{\text{adv}} \tag{3}$$

Then we have the cyclical consistency loss, which features the cycleGAN network:

$$\mathcal{L}_{cc} = \mathbb{E}_{\mathcal{I},\mathcal{I}^a}\left[\|\tilde{y} - y\|_1\right] + \mathbb{E}_{\mathcal{I},\mathcal{I}^a}\left[\|\tilde{x}^a - x^a\|_1\right] \tag{4}$$

We add a new loss function called artifact consistency loss to make sure that artifact removed from x^a has the similar shape as the artifact added to y.

$$\mathcal{L}_{\text{art}} = \mathbb{E}_{\mathcal{I},\mathcal{I}^a}\left[\|(x^a - \hat{x}) - (\hat{y}^a - y)\|_1\right] \tag{5}$$

The purpose is to make sure that generator B, which contains $E_{\mathcal{I}}$, $G_{\mathcal{I}^a}$ and $E^a_{\mathcal{I}^a}$ can get the proper training so that the artifact removed from x^a can be properly added to y.

5 Experiments

5.1 Artifact reduction

5.1.1 CycleGAN without FPN

Some parameter settings:
batch size =2;epoch=20

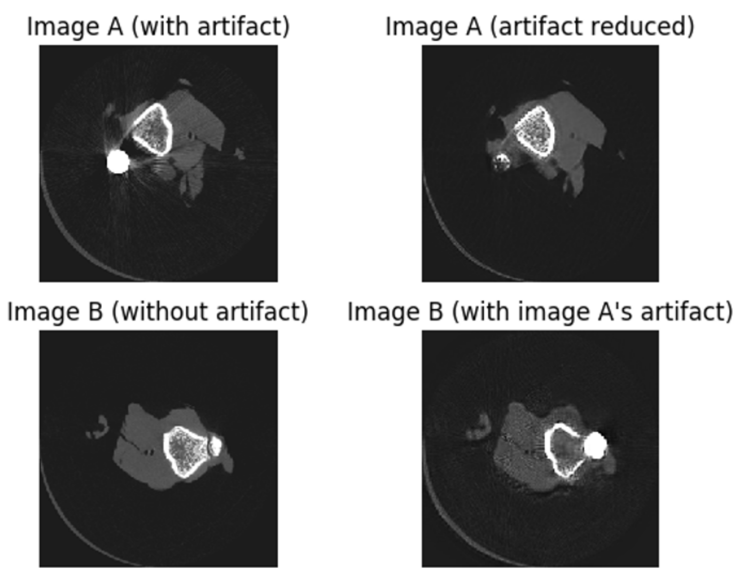

Figure 5: Reconstruction without FPN

Obviously, there are still enough metal and streaked artifact remained, which means the generator doesn't learn the distribution of artifact data correctly. The traditional cycleGAN has the space of improvement.

5.1.2 CycleGAN with FPN

Firstly, we trained the network with 20 epoch and batch size equals to 2. The results are as follows:

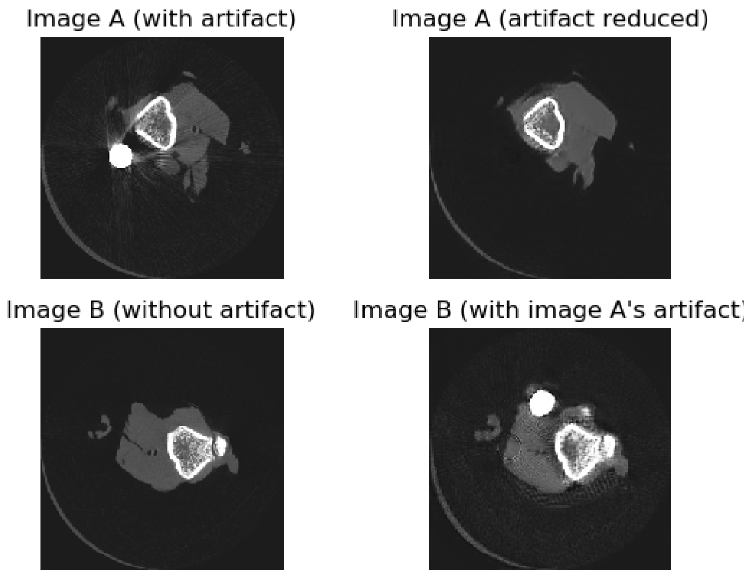

Figure 6: reconstruction image

The figure in the right corner of the fig.3 above demonstrates that the artifact has been removed. So the model outperforms the one without FPN structure. However, some anatomy information are removed simultaneously.

5.2 Image quality assessment

We apply SSIM(structure similarity) as well as PSNR(peak signal to noise ratio) metric to assess the quality of images quantitatively.

	SSIM	PSNR
vanilla cycleGAN	79.4	20.21
cycleGAN with FPN	84.3	20.1

Table 1: metric comparison

In terms of SSIM, cycleGAN with FPN has the better performance. However, there isn't much difference in terms of PSNR.

6 Conclusions

In this article, we utilize ADN's work as our source of reference to build our cycleGAN model. We find that Feature pyramid structure can greatly help the generator to learn the artifact features and add them to the artifact-free picture. After that, artifact-consistency loss makes sure that artifact can be ripped off from the original picture. The visual and quantitative result shows that our method has advantage over the normal cycleGAN network.

References

[1] J. Wang, Y. Zhao, J. H. Noble, and B. M. Dawant, "Conditional Generative Adversarial Networks for Metal Artifact Reduction in CT Images of the Ear," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, ser. Lecture Notes in Computer Science, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham: Springer International Publishing, 2018, pp. 3–11.

[2] Y. Zhang and H. Yu, "Convolutional Neural Network Based Metal Artifact Reduction in X-Ray Computed Tomography," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1370–1381, Jun. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8331163/

[3] H. Liao, W.-A. Lin, S. K. Zhou, and J. Luo, "ADN: Artifact Disentanglement Network for Unsupervised Metal Artifact Reduction," *IEEE Transactions on Medical Imaging*, vol. 39, no. 3, pp. 634–643, Mar. 2020, arXiv: 1908.01104. [Online]. Available: http://arxiv.org/abs/1908.01104

[4] Y. Lyu, J. Fu, C. Peng, and S. K. Zhou, "U-DuDoNet: Unpaired dual-domain network for CT metal artifact reduction," *arXiv:2103.04552 [cs, eess]*, Mar. 2021, arXiv: 2103.04552 version: 1. [Online]. Available: http://arxiv.org/abs/2103.04552

[5] W.-A. Lin, H. Liao, C. Peng, X. Sun, J. Zhang, J. Luo, R. Chellappa, and S. K. Zhou, "DuDoNet: Dual Domain Network for CT Metal Artifact Reduction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 10 504–10 513. [Online]. Available: https://ieeexplore.ieee.org/document/8953298/

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *arXiv:1703.10593 [cs]*, Aug. 2020, arXiv: 1703.10593. [Online]. Available: http://arxiv.org/abs/1703.10593

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *arXiv:1612.03144 [cs]*, Apr. 2017, arXiv: 1612.03144. [Online]. Available: http://arxiv.org/abs/1612.03144

A Appendix

The structure of encoder and decoder is copied from ADN. And we modified some parameters to get better performance:

Network	Block/Layer	Count	Ch.	Kernel	Stride	Pad.
$E_{\mathcal{I}}$ / $E_{\mathcal{I}^a}$	down	1	32	7	1	3
	down.	1	64	4	2	1
	down.	1	128	4	2	1
	residual	4	128	3	1	1
E_a	down.	1	32	7	1	3
	down.	1	64	4	2	1
	down.	1	128	4	2	1
$G_{\mathcal{I}}$	residual	4	128	3	1	1
	up.	1	64	5	1	2
	up.	1	32	5	1	2
	final	1	1	7	1	3
$G_{\mathcal{I}^a}$	residual	4	128	3	1	1
	merge	1	128	1	1	0
	up.	1	64	5	1	2
	merge	1	64	1	1	0
	up.	1	32	5	1	2
	merge	1	32	1	1	0
	final	1	1	7	1	3
$D_{\mathcal{I}}$ / $D_{\mathcal{I}^a}$	conv	1	32	4	2	1
	relu	1	-	-	-	-
	down.	1	64	4	2	1
	down.	1	128	4	1	1
	conv	1	1	4	1	1

Figure 7: network configuration