CT Image Reconstruction and Optimization with Unsupervised Deep Learning

Yanchuan Yin[1,a]

[1]Beijing Forestry University, Beijing, China
[a]yanchuanyin@163.com

Abstract The quality of computed tomography (CT) images is often affected by a variety of factors, such as the patient's movement or hardening of the beam, and the fact that high doses of radiation cannot be used in consideration of human health. Traditional CT image reconstruction methods and learn-based CT image reconstruction techniques face challenges such as missing edge information, ring artifacts and obvious data dependence. We applied the unsupervised learning model to the field of medical images, completed the denoising, high-resolution, inpainting, and restoration tasks of CT images, and optimized the original model. The model parameters are adjusted to improve the performance of the model, and the adjustment results are evaluated by data analysis and image visualization[1].

Keywords deep image prior, CT image construction, parameter optimization

1 Introduction

1.1. Background

CT image quality is often compromised by various factors, including damage, oil stains, handwritten notes, patient movement, and cardiac rhythm, necessitating repairs. Additionally, reducing CT scan doses to minimize radiation exposure impacts image clarity. Deep learning has achieved superior reconstruction quality for solving inverse problems[1] but requires extensive training data, raising concerns about patient exposure to high X-ray doses. Meanwhile, Traditional methods like filtered back projection (FBP) and iterative reconstruction can lead to edge information loss and central ring artifacts[3]. Learning-based techniques, while data-intensive, risk overfitting and performance decline without ample labeled CT data.

1.2. Challenge

(1) Adjusting model parameters and hyperparameters is constrained：excessive tweaking can lead to overfitting or high computational costs. The adjustment process lacks a fixed pattern, necessitating repeated exploration for optimal settings.

(2) Model optimization via backpropagation seeks network weights but may require numerous iterations to reach the global optimum, with a risk of settling into local minima.

(3) This study employs CT images as input, which, due to their medical nature, demand high information fidelity and accuracy. Prior research seldom uses CT images, resulting in lower repair accuracy and challenging parameter tuning.

[1] https://github.com/YYC-CHUAN/dip-CTuse.git

1.3. Contribution

(1) In the field of Medical image application, using damaged MR, US and PET/CT images as input, and using the unsupervised learning model of image reconstruction based on the internal understanding of natural images, the recovered CT images can be outputed in seven tasks, including denoising, high resolution, natural preimage, restoration and so on.

(2) Algorithm optimization of the original model is carried out. The performance of the model is enhanced by adjusting the model parameters. LiTS liver tumor data set and MMNHS heart segmentation data set were used for data evaluation. We changed model parameters and hyperparameters and compared the model performance with curve ICONS of the MSE and PSNR values. Meanwhile, subjective quality of visual perception was used as the standard for testing.

2. Related work

Deep learning methods used in CT image reconstruction, such as denoising, super-resolution, inpainting, and enhancement, have shown great promise but also face limitations. We completed four tasks which mentioned: (1) Denoising: Denoising techniques[6][7] like DIP may overfit data or require extensive high-quality scans, making them costly and dependent on prior models. (2) Super-resolution： Super-resolution methods[8][9] demand paired high- and low-resolution images for training, which limits their real-world applicability and can be affected by noise. (3) Inpainting: Inpainting models[10][11] struggle with data reliance and may produce less natural images due to a lack of detailed metal edges. (4) Enhancement： Enhancement algorithms[12][13], while improving contrast, have limited noise suppression capabilities and can exacerbate existing flaws in low-resolution images. Future research should focus on developing robust, data-efficient, and noise-resistant methods to improve CT image quality.

3. Method

The parametric function of the deep generator network is $x = f_\theta(z)$.It maps the coding vector z to the image *x*. By converting the simple distribution p(z) of z to the complex distribution p(x) of image x, the simple input transformation simulates a more complex and rich image. Often deep generator networks need to be trained with a lot of real data to adjust the value of θ to encode knowledge within the distribution *p(x)* into θ. However, in the method used in this paper, the information of the image distribution *p(x)* is contained in the network structure, and no training on the parameter θ in the network is required.

3.1. Network overview

In the method proposed in this paper, the size of the image tensor is $x \in R^{3\times H\times W}$ (H means the height of tensor, W means the width) , the size of the coding vector is fixed as $Z \in R^{C'\times H'\times W'}$, and the structure of the neural network is parametrization $x = f_\theta(z)$. The neural network iterates to obtain θ value according to the input image through algorithms such as gradient descent. In each iteration, the randomly initialized coding vector *z* performs output image according to θ, and when θ reaches the local minimum point, the output image is completed. It can be said that the network will map to the image *x* based on parameters θ generated by x_0, comprising weights and filter deviations in the network. The network itself has a standard structure and alternates filtering operations such as linear convolution, upsampling and non-linear activation functions.

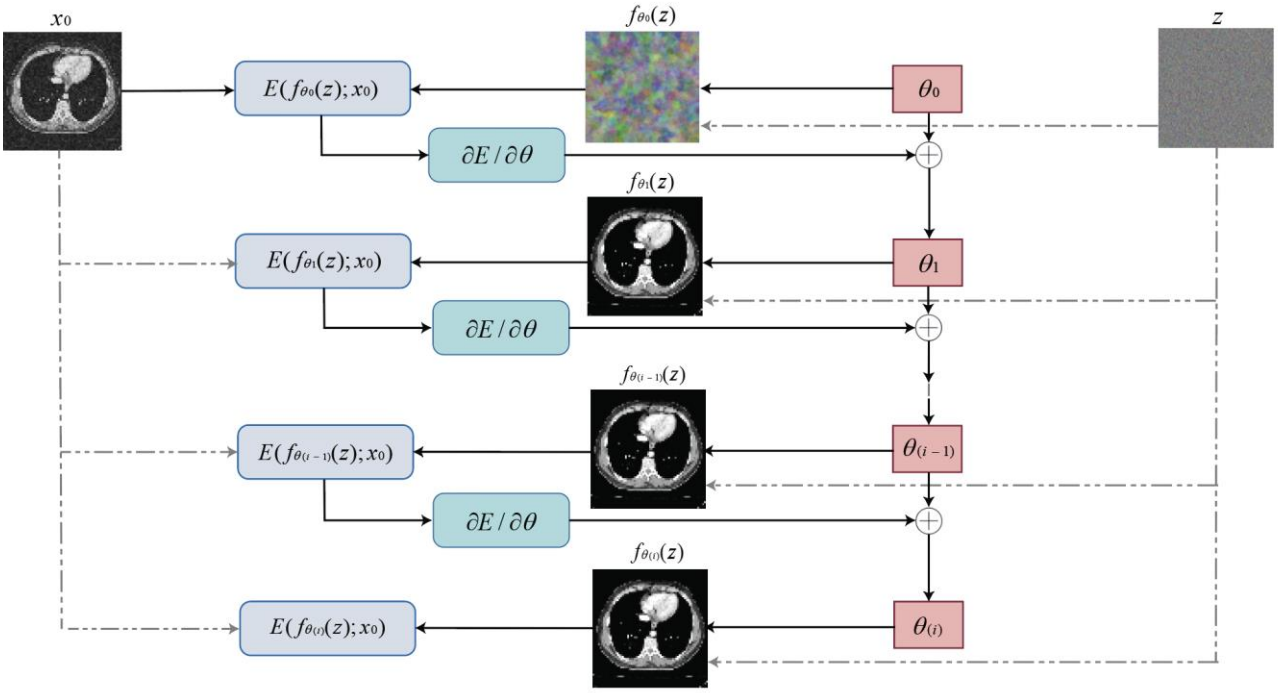

x_0
$f_{\theta_0}(z)$
z
$E(f_{\theta_0}(z); x_0)$
θ_0
$\partial E / \partial \theta$
$f_{\theta_1}(z)$
$E(f_{\theta_1}(z); x_0)$
θ_1
$\partial E / \partial \theta$
$f_{\theta(i-1)}(z)$
$E(f_{\theta(i-1)}(z); x_0)$
$\theta_{(i-1)}$
$\partial E / \partial \theta$
$f_{\theta(i)}(z)$
$E(f_{\theta(i)}(z); x_0)$
$\theta_{(i)}$

Figure 1:Network framework in image restoration.

3.2. Process

We output the images by the following steps:

(1) Randomly initialize the network weights θ to generate an initial image $x = f_\theta(z)$ from a fixed random 3D tensor z.

(2) Update θ using gradient descent and other algorithms based on the energy function $E(f_\theta(z); x_0)$, where x_0 is the input image.

(3) Iteratively update θ with the energy term $E(f_\theta(z); x_0)$ using the same random 3D vector z until convergence.

(4) Output the recovered image $f_\theta(z)$ when θ reaches a local optimum, either when $E(f_\theta(z); x_0) = 0$ or when the optimal stopping point is reached.

Details are included in the appendix.

3.3. Network with resistance to high noise

Although almost all images can be adapted to the network, the adaptation degree of low noise images and high noise images is different, the essence of which is that the network architecture has selective retention of information. The neural network will resist the results that do not conform to the ground truth image, and will decline in the direction of the ground truth, and the decline efficiency of the natural image is higher.

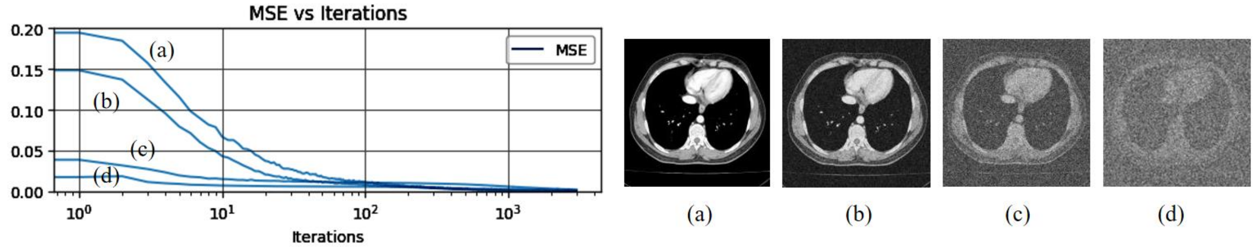


Fig2：Learning curves for the reconstruction task using CT images with different levels of Gaussian noise.

In order to quantify the influence of different image noise intensity on the network reconstruction effect, we compared the distance L^2 between the generated image and the input image x_0:

Figure 2 shows the energy $E(x;x_0)$ The iterative process under the selection of four different images: (1) a natural image,(2) the same image with Gaussian noise intensity of var=0.01, (3) the same image with Gaussian noise intensity of var=1, (4) the same image with Gaussian noise intensity of var=10.

From the image it can be seen that the network optimizes the original image and the natural image with low noise intensity faster, while it has obvious resistance to strong noise. This reflects the characteristics of parameterization: the retention of signal information is preferred over noise. According to this conclusion, in the actual image reconstruction problem, we limit the number of iterations to ensure that the model learning is focused on the signal information to reduce the sensitivity to noise, and use the network internal regularization to avoid noise overfitting.

3.4. Details in applications

(1) Denoising: Under the assumption of blind denoising, we use the same algorithm as equations 3 and 4. Given a CT image degraded due to complex and unknown compression artifacts, the restored clean image is output after substituting the minimize $\theta*$.

(2) Super resolution: Super resolution is a problem of obtaining a higher resolution image from the reverse of the resolution image, we replace the energy term with

$E(x;x_0) =\parallel d(x) - {x_0}^2 \parallel$, where $d(\cdot)$ is a subsampler calculated by minimizing the distance between the desired high-resolution image and the original image and using the subsampler in reverse. Regularization is used at the same time to pick out the output image that best matches the expectation.

(3) Inpainting: We replace the energy term with $E(x;x_0) =\parallel (x - x_0) \odot m \parallel^2$ to measure the difference between the current image x and the missing part of the original image x_0, and transform the optimization problem into the solution of the latent variable z, using z to indirectly affect the value of x to introduce the prior knowledge constraint solution.

(4) Restoration: Restoration considers image restoration drawn according to a mask randomly sampled from a binary Bernoulli distribution. We sample a mask and randomly drop the pixels by 50%. In this case, the missing part of the image is repaired.

4. Experiment

In this experiment, we used a T4 GPU (memory size of 15.0GB), system RAM of 12.7GB, and disk size of 112.6GB. The medical image data used came from two datasets of AI Studio website: CHAOS Multi-organ Segmentation dataset and TCGA-LIHC Liver CT. Web site https://aistudio.baidu.com/datasetdetail/23864 and https://aistudio.baidu.com/datasetdetail/37439 , respectively. The CHAOS dataset includes 40 liver/spleen/kidney CT and MRI scans with segmentation labels, while the TCGA-LIHC Liver CT dataset comprises clinical images.

We applied the medical image data to the unsupervised deep learning model and optimized the algorithm, adjusted the learning rate, no.scales and other parameters to improve the model performance, and obtained the optimal optimization result by comparing the final PSNR value of different experimental parameters and the visual perception analysis of the output image.

We completed a total of four image reconstruction tasks and summarized the results into PSNR line chart and output result comparison chart. However, due to space limitations, we put two tasks in the appendix and only described two tasks in the main body.

4.1. Denoising

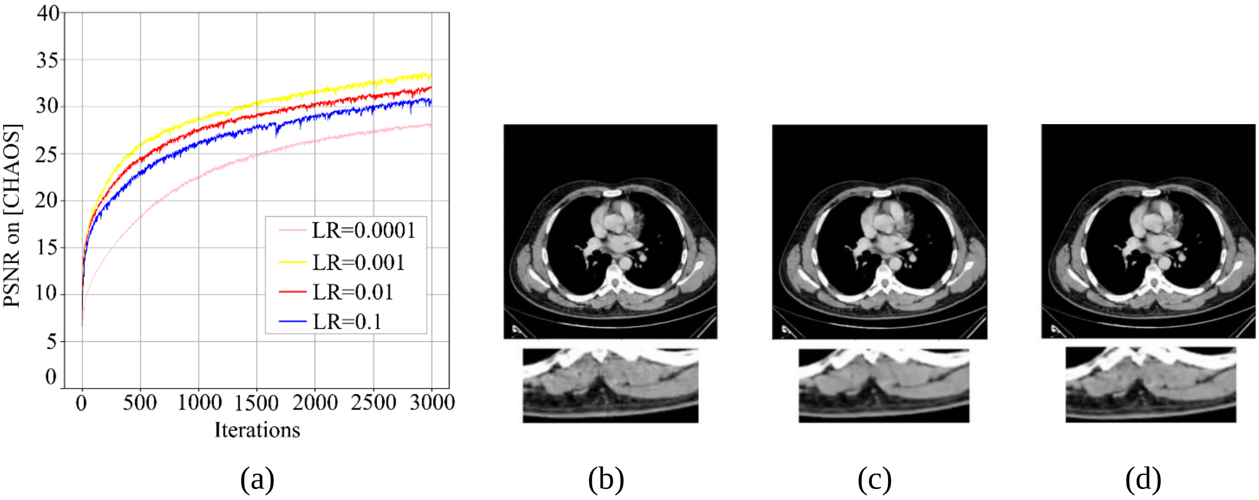


Figure 3: (a) performance of denoising comparing on different learning rates; (b) the original image; (c) output image when learning rate = 0.001;(d) output image when no.scale = 3.

In denoising application, we adjusted the values of learning rate {0.0001,0.001,0.01,0.1} and no.scales {3,4,5,6,7,} for CHAOS data set by means of control variables, and evaluated the quality of output images with PSNR values.

In the process of adjusting the learning rate, we found that the learning rate of the original model was 0.01, and the output image displayed the best quality, even better than the original image, when it was exponentially reduced to 0.001, while the output image quality was not as good as the original when it was reduced to 0.001 or increased to 0.1.

In the process of adjusting the values of no.scales , we found the result was similar with the one in adjusting learning rate,where the output image quality would be the best when the no.scales was reduced to 3.

4.2. Inpainting

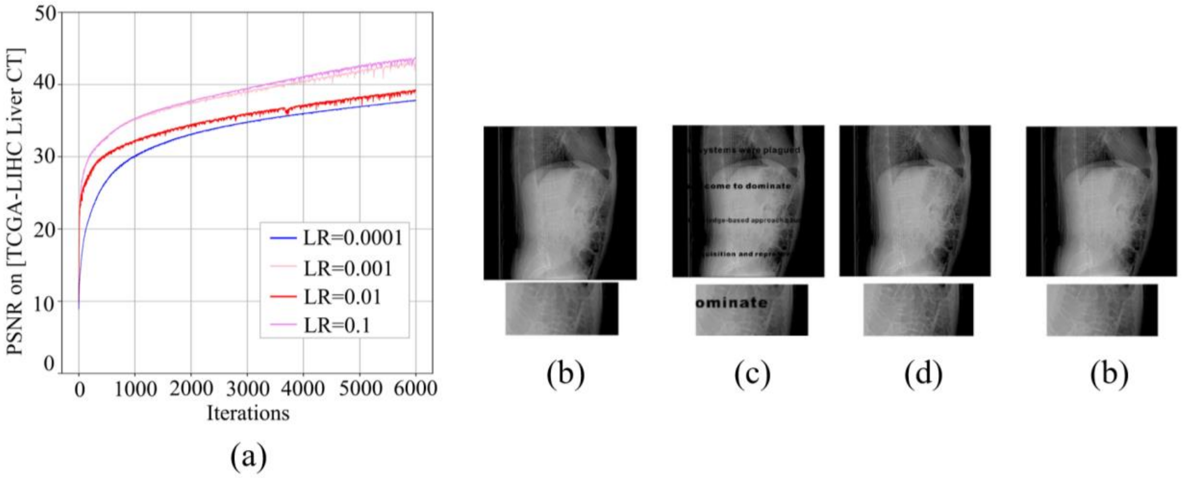


Fig 4: (a) Performance of inpainting comprising on different learning rate. (b) the original image; (c) the corrupted image; (d) output image when learning rate = 0.001； (e) output image when no.channels= 4.

In the inpainting application, we adjusted the values of the learning rate {0.0001,0.001,0.01,0.1} and no.channels (3,4,5,6,7} for the TCGA-LIHC Liver CT dataset using the same method as denoising.

We find that the results of the adjusting are similar to the previous application, where the model works best when the learning rate was exponentially expanded to 0.1 from 0.01, which is the original value.

In the process of adjusting no.channels in inpainting application, we found that the original model no.channels was 5, and the output image displayed the best quality when it was reduced to 4, followed by the effect when it was expanded to 6 and 7, both of which were better than the original output results, while the output image was poor when it was reduced to 3.

5. Conclusion

In this paper, we apply CT images to an unsupervised deep learning image reconstruction model and optimize the reconstruction results by changing the model parameters and hyperparameters. In the future work, we can continue to optimize the restoration part and enhance the application range of CT image restoration.

6. REFERENCE

[1] *Baguer, D. O., Leuschner, J., & Schmidt, M. (2020). Computed tomography reconstruction using deep image prior and learned reconstruction methods. Inverse Problems, 36(9), 094004.*

[2] *Cui, J., Gong, K., Guo, N., Wu, C., Meng, X., Kim, K., ... & Li, Q. (2019). PET image denoising using unsupervised deep learning. European journal of nuclear medicine and molecular imaging, 46, 2780-2789.*

[3] *Diwakar, M., & Kumar, M. (2018). A review on CT image noise and its denoising. Biomedical Signal Processing and Control, 42, 73-88.*

[4] *Gondara, L. (2016, December). Medical image denoising using convolutional denoising autoencoders. In 2016 IEEE 16th international conference on data mining workshops (ICDMW) (pp. 241-246). IEEE.*

[5] *Kaur, P., Singh, G., & Kaur, P. (2018). A review of denoising medical images using machine learning approaches. Current medical imaging, 14(5), 675-685.*

[6] *Cui, J., Gong, K., Guo, N., Wu, C., Meng, X., Kim, K., ... & Li, Q. (2019). PET image denoising using unsupervised deep learning. European journal of nuclear medicine and molecular imaging, 46, 2780-2789.*

[7] *Baguer, D. O., Leuschner, J., & Schmidt, M. (2020). Computed tomography reconstruction using deep image prior and learned reconstruction methods. Inverse Problems, 36(9), 094004.*

[8] *Umehara, K., Ota, J., & Ishida, T. (2018). Application of super-resolution convolutional neural network for enhancing image resolution in chest CT. Journal of digital imaging, 31, 441-450.*

[9] *Park, J., Hwang, D., Kim, K. Y., Kang, S. K., Kim, Y. K., & Lee, J. S. (2018). Computed tomography super-resolution using deep convolutional neural network. Physics in Medicine & Biology, 63(14), 145011.*

[10] *Park, J., Hwang, D., Kim, K. Y., Kang, S. K., Kim, Y. K., & Lee, J. S. (2018). Computed tomography super-resolution using deep convolutional neural network. Physics in Medicine & Biology, 63(14), 145011.*

[11] *Peng, C., Li, B., Li, M., Wang, H., Zhao, Z., Qiu, B., & Chen, D. Z. (2020). An irregular metal trace inpainting network for x-ray CT metal artifact reduction. Medical Physics, 47(9), 4087-4100.*

[12] *Salem, N., Malik, H., & Shams, A. (2019). Medical image enhancement based on histogram algorithms. Procedia Computer Science, 163, 300-311.*

[13] *Yamashita, K., & Markov, K. (2020). Medical image enhancement using super resolution methods. In Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part V 20 (pp. 496-508). Springer International Publishing.*

[14] *CHAOS Multi-organ Segmentation dataset: https://aistudio.baidu.com/datasetdetail/23864*

[15] *TCGA-LIHC Liver CT: https://aistudio.baidu.com/datasetdetail/37439*

Appendix

A Method step details

A.a Step 1: Initialize parameter θ randomly

At the beginning of the first iteration, the network randomly generates weight θ, starting at θ_0, which is mapped to the image $x = f_\theta(z)$, where z is a fixed random initialization 3D tensor that will be used for image generation in each iteration, and parameterized f is a neural network with parameter θ.

A.b Step 2: Update parameter θ

The image generated by the first iteration and the input image are used to calculate $E(f_\theta(z); x_0)$ and update the parameter θ by gradient descent and other algorithms.

In this work, we change the distribution that deals with the input image explicitly in the calculation to a more optimized and indirect energy minimization problem, as follows:

$$x^* = \underset{x}{argmin} E(x; x_0) + R(x) \tag{1}$$

Where x_0 is a noisy/low-resolution/occluded image, $E(x; x_0)$ represents the task dependent part based on the original or noisy image x_0. The data item $\underset{x}{argmin} E(x; x_0)$changes according to the applied task. $R(x)$ is a regularizer that is independent of specific applications because it captures the general regularity of natural images, such as TV regularization,which encourages images to contain uniform regions, but much research has gone into designing and learning good regularizers.

We further drop $R(x)$which captures the generic regularity of images explicitly to utilize the implicit prior knowledge of neural network parameterization, as follows:

$$\theta^* = \underset{\theta}{argmin} E(f_\theta(z); x_0), x^* = f_\theta \cdot (z) \tag{2}$$

In this computation method, we redefine the parameter optimization goal to find the optimal value of θ , and when given x_0, the model is able to minimize the expected error $E(f_\theta(z); x_0)$ through an optimizer (such as gradient descent) to obtain the local minimum point of θ, where the parameter θ starts as a random initialization parameter. There is no additional annotation or input in the entire image reconstruction algorithm, which completely relies on the observation data x_0. The only empirical information available to the restoration process is the noisy image x_0.

At the same time, the untrained network is able to capture low-level statistics of natural images (such as the local and translation invariant nature of convolutions, as well as pixel neighborhood relationships at different scales), suggesting that the interactions between weights and layers within the network can capture the underlying structure of the image, rather than simply replicating the noise in the image. This will be very useful for the application of this method to image reconstruction problems.

A.c Step 3: Repeated iteration

In each subsequent iteration, the input image x_0 is used for each energy term $E(f_\theta(z); x_0)$ is calculated to update the parameter θ, and the 3D vector z generated by random initialization is used to generate the image x on each iteration update.

A.d Step 4: Image output

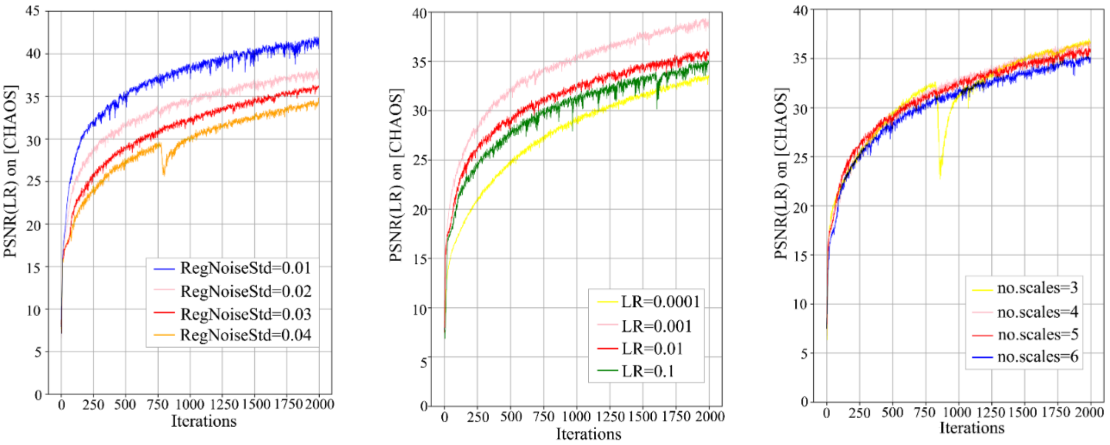
When θ reaches the local optimal value, $E(f_\theta(z);x_0) = 0$ or $E(f_\theta(z);x_0)$ When the optimal stopping point is reached, the iteration is stopped and the recovered image $f_\theta(z)$ is output. The two cases are:

(1) the ground-truth solution x_{gt} belongs to the space surface of x without energy : $E(x;x_0) = 0$ The method used in this paper can realize that the output image is close to the ground truth by re-parameterizing and adjusting the optimization trajectory.

(2) the ground-truth has non-zero $E(x_{gt};x_0) > 0$. At this time, even if the network iterates many times so that $E(x;x_0) = 0$, it cannot get close to the ground-truth image. In general, the optimization path will pass near the xgt (optimal stopping point) and stop early, where the output is closer to the ground-truth image.

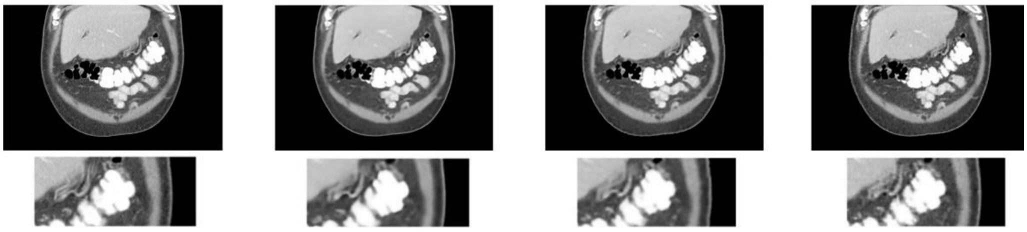
B Experiment result

B.a Super-resolution



(a) (b) (c)

Fig 5：(a) Performance of inpainting comprising on different RegNoiseStd.(b) Performance of inpainting comprising on different learning rates.(c) Performance of inpainting comprising on different no.scales.



(a) (b) (c) (d)

Figure 6：(a) the original image; (b) high-resolution output image when learning rate is 0.001; (c) high-resolution output image when no.scales is 37.58; (d) high-resolution output image when RegNoiseStd is 0.01.

In super-resolution application, we adjusted the values of learning rate {0.0001, 0.001, 0.01, 0.1} , no.scales {3, 4, 5, 6} and RegNoiseStd {0.01, 0.02, 0.03, 0.04} for CHAOS data set using the same method, and evaluated the quality of output images with PSNR(HR) and PSNR(LR) values.

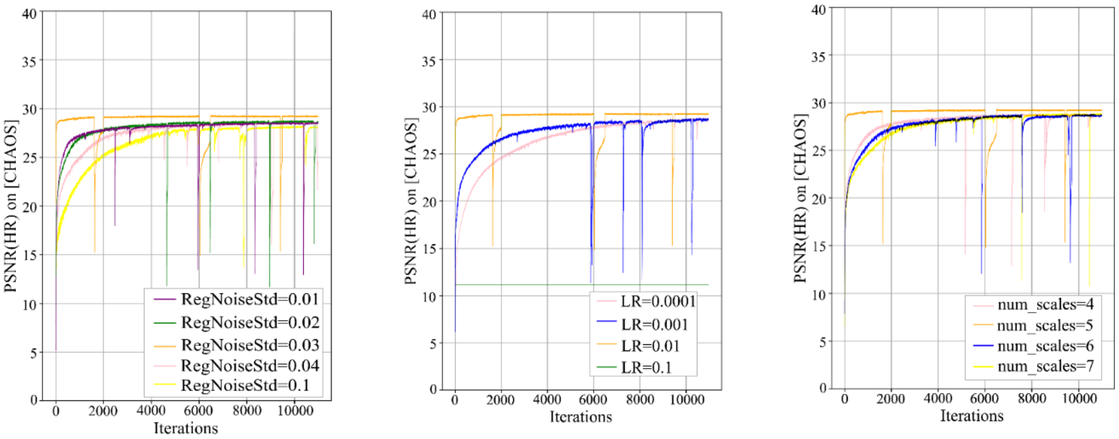
PSNR(HR) is the PSNR value calculated between the high-resolution (High Resolution, HR) image processed by the neural network model and the actual high-resolution reference image, while PSNR(LR) is the PSNR value calculated between the low-resolution (Low Resolution, LR) input image processed by the same network and the original low-resolution image.

In the process of adjusting the values of RegNoiseStd, we find that the results of the adjusting are similar to the previous application, where the model works best when the RegNoiseStd was reduced to 0.01 from 0.03, which is the original value. However,for low-resolution images, the effect of model optimization is much more obvious.

In the process of adjusting the learning rate, we found that the learning rate of the original model was 0.01, and the output image displayed the best quality, even better than the original image, when it was exponentially reduced to 0.001, while the output image quality was not as good as the original when it was reduced to 0.0001.

In the process of adjusting the values of no.scales, we found that the original model no.scales was 5, and when the no.scales was reduced to 4, the output image quality would be the best, while when the no.scales was reduced to 3 or increased to 6 , the output image quality is not as good as that of 5.

B.b Restoration



(a) (b) (c)

Fig 7：(a) Performance of inpainting comprising on different RegNoiseStd.(b) Performance of inpainting comprising on different learning rates.(c) Performance of inpainting comprising on different no.scales.

In restoration application, we adjusted the values of learning rate {0.0001, 0.001, 0.01, 0.1}, no.scales {4, 5, 6, 7} and RegNoiseStd {0.01, 0.02, 0.03, 0.04} for CHAOS data set by means of control variables, and evaluated the quality of output images with PSNR values.

However, we found that the performance with original parameters is the best, in which the learning rate is 0.001,the no.scales is 5 and the RegNoiseStd is 0.03.

The performance with original parameters is the best in restoration adjustment.