Standard Deviation Noise Maps of CT Scans using UNet and UNet-based Architectures

Avantika Kothandaraman, Caiwei Zhang, Long Chen
avantk@bu.edu, magzcw@bu.edu, lchen122@bu.edu

1. Task

The goal of this project is to develop a deep learning model that can accurately estimate the standard deviation (STD) map of noise in CT scans. This model will take in raw CT scan data like those shown in Figure 1, process it through UNet and UNet-based architectures, and output a detailed STD map, which reflects the localized noise levels across the scan.

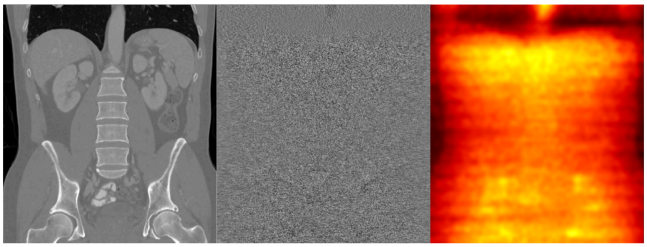


Figure 1: Left: Original CT scan signal; Center: Isolated noise component; Right: Standard deviation map

The main challenge lies in accurately modeling noise without compromising the integrity of anatomical details in the CT scans. Deep learning models, particularly those involving complex architectures like UNet, must preserve spatial relationships within the images to ensure that medical insights derived from the scans are not distorted. The task demands a careful balance between noise estimation and the preservation of essential medical information, posing a significant test of the model's ability to handle real-world medical imaging data.

2. Related Work

Huber et al. developed a U-Net convolutional neural network with mean-square-error loss called SILVER to estimate the local noise level within each region of a CT image [1]. The noise maps were generated by calculating the pixel-wise standard deviation on 120,000 images from three anthropomorphic phantoms representing the chest, head, and pelvis. These maps served as training targets. The trained SILVER framework was evaluated using both phantom and patient CT images, showing high accuracy. However, the limitations of this study include the inability to calculate the noise level within non-uniform patient regions, and the model is also dependent on the ability to include similar features within the anthropomorphic phantom training data.

In [2], Ronneberger et al. introduced UNet, a pioneering deep learning architecture specifically designed for medical image segmentation. This Convolutional Neural Network (CNN) derives its name from its distinctive "U" shape, created by its encoder-decoder structure. The encoder, performing downsampling through a series of convolutions and max pooling, funnels into a bottleneck layer. This layer then transitions into the upsampling decoder, which restores spatial details lost during downsampling. A critical innovation of the UNet is its use of skip connections, which facilitate the transfer of feature data directly across from the encoder to the decoder, preserving intricate image details.

In [3], a residual U-Net structure, named RatUNet, modifies the traditional U-Net by integrating residual blocks from ResNet to deepen the network and prevent performance saturation. It also improves the downsampling and upsampling methods to better extract and reconstruct image features. Additionally, it uses depthwise and polarized self-attention mechanisms to enhance edge detail and overall image quality, which are useful for accurate noise reduction. The paper shows that RatUNet is more efficient and has better performance than a number of recent state-of-the-art methods. However, the enhanced depth and complexity of the network might also make it prone to overfitting, especially when trained on limited or overly homogeneous datasets.

Both studies [4] and [5] underscore the utility of patch-based approaches within their respective fields. Alkinani et al.'s research demonstrates that patch-based methods significantly outperform pixel-based alternatives in noise reduction by exploiting the redundancy and similarity among patches to enhance denoising quality. However, as discussed in the second study, the interaction between patch size

and batch size is flexible for optimization, so precise adjustments are essential for maximizing performance.

3. Approach

The project was first begun by exploring the dataset and the images in it. By exploring the images, one can understand the range of intensities, shapes and sizes, 3D nature, and how the images can be transformed into patches. For initial data exploration and analysis, basic libraries like Torch, Numpy, Scikit-Learn, and Matplotlib were used. To handle specific data format requirements, a custom dataset class was developed, where the images were loaded with Python's NRRD library. This choice was driven by the need to maintain the integrity of the original medical imaging data during the loading process. Our CustomDataset class efficiently manages large volumes of NRRD image files. Upon loading, each image undergoes a transformation into a 2D representation. Then, all the images in the given dataset were converted to PyTorch tensors for further analysis.

To augment the dataset and simulate real-world variabilities in imaging, random noise (which was provided in the dataset) was added to the main CT signal images. This procedure generates a new set of images, termed CT_generated, by adding scaled noise derived from corresponding noise images in the dataset (Figure 2 shows examples). This generated image and the STD map from the dataset were loaded into the custom dataset.

$$ct_{generated} = ct_{signal} + (k \times ct_{noise}) \ \ where \ k = rand[0,5]$$

Equation 1: Generating ct_generated

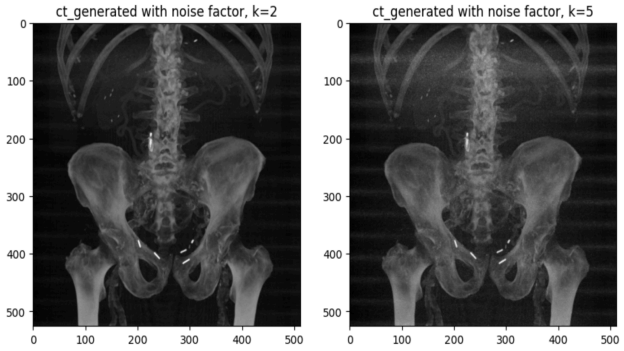


Figure 2: (Left) ct_generated with noise factor, k=2 (Right) ct_generated with noise factor, k=5

For effective model training, the Patchify library was used to segment both CT_generated and the standard deviation (STD) maps into non-overlapping 2D coronal patches of size 64x64. The reason for generating patches was to increase the amount of data available for training and to reduce the computational complexity that is often seen during the training of complex Deep Learning models like UNet and RatUNet.

3.1. UNet

The first deep learning architecture explored in this project is the UNet, which was originally developed in 2015 specifically for medical image segmentation [2]. In contrast to its conventional application, here, the goal is to estimate a standard deviation map of the noise in CT scans. The model takes CT_generated images as input and the target output is the STD map, which is the calculated standard deviation noise map provided in the dataset. Therefore, in this project, the UNet is used in a regression task as opposed to its traditional use in segmentation tasks.

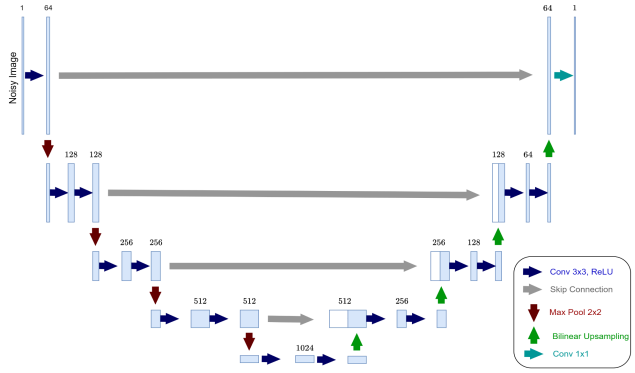


Figure 3: UNet architecture

This traditional U-Net architecture, a special type of Convolutional Neural Network that is structured as a U-shaped model, was used in this project, as illustrated in Figure 3. This architecture utilizes five encoder-decoder layers: 64, 128, 256, 512, and 1024, with the 1024-channel layer serving as the bottleneck. The encoder serves as a downsampling mechanism to extract important features, while the decoder performs upsampling to reconstruct spatial information that may have been lost during encoding.

During encoding, the input image is put through a series of convolutions and max pooling layers with ReLU activation. The decoder reduces the number of feature channels while concatenating the layers and the channels in them. Skip connections help preserve the fine details and enable the flow of data between the encoder and decoder layers. The convolutional kernel used in this UNet is a 2x2 kernel. The loss function used in this case is Average Relative Error, which provides a measure of the deviation between the

predicted standard deviation maps and the ground truth.

$$LOSS = \frac{1}{n}\sum_{i=0}^{n}\frac{\left|y_i - \widehat{y_i}\right|}{y_i + \varepsilon}, \text{ where } \varepsilon = 1e^{-8},$$

y_i is the target, $\widehat{y_i}$ is the output from model.

Equation 2: Loss Function - Average Relative Error

Batch Normalization was done to normalize the inputs and to ensure faster training. Dropout Regularization was done to ensure that there was no overfitting during training. Here, for UNet, the dropout rate was chosen to be 0.1, meaning that each neuron in the layer has a 10% probability of being dropped from the training process, as in, set to zero. The target image fed into the model is a calculated standard noise map of the input images. The output of this model would be the predicted standard deviation noise map for the given input image.

3.2. RatUNet

Given that the dataset comprises signals, noise, and STD maps, the primary research question focuses on predicting the STD map from the signals and noise. However, the inverse problem of estimating the signal from a noisy image presents another intriguing question. This bidirectional analysis led to the exploration of the residual UNet-based attention mechanism (RatUNet) architecture, originally designed for image denoising. In its standard form, RatUNet outputs denoised images by addressing and removing noise characteristics in its final layer. We hypothesized that with appropriate modifications from Zhang et al.'s application of RatUNet, a similar architecture could be adept at predicting STD maps. To this end, we adapted the architecture to better align with the specific requirements of STD map prediction, and the modified RatUNet architecture is visually depicted in Figure 5.

The key modifications implemented are outlined as follows:

a. The single convolution layer was replaced with a sequence of two convolutions, each followed by batch normalization, ReLU activation, and a dropout mechanism. This enhancement is intended to capture more complex features and introduce regularization at an early stage, which could be crucial for handling diverse scales of noise and signal variations in the data.

b. Originally, the network was configured to subtract the predicted noise from the input to yield a cleaned image. This approach was modified to focus directly on generating the STD map as the output. This shift aligns with the objective of directly learning the variability represented by the STD map, rather than enhancing or correcting specific features of the input image.

c. Additional batch normalizations were integrated within the residual blocks. These adjustments aim to improve training stability and performance by ensuring more consistent activation distributions throughout the network [6]. The architecture is illustrated in Figure 4.

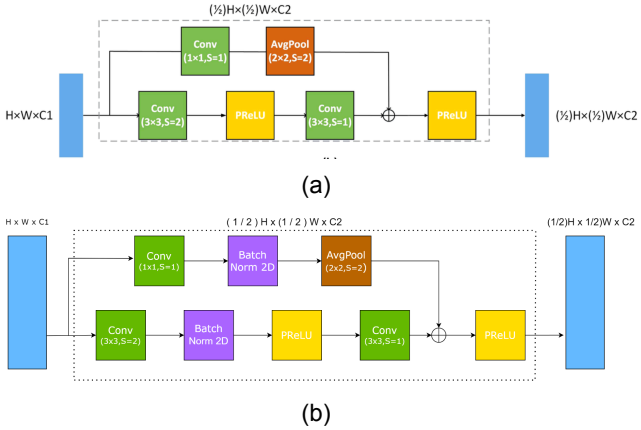


Figure 4: Residual block architecture. (a) The residual block used in the original RatUNet [2]. (b) The modified residual block in our model. (s is stride).

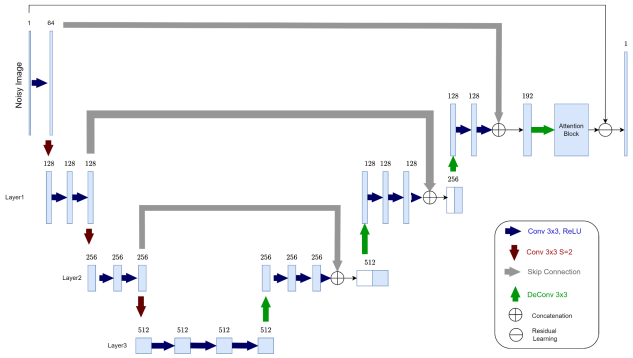


Figure 5: RatUNet architecture

With modifications to the original RatUNet, experiments were conducted similar to those performed with the U-Net model. This includes testing dropout rates of 0.0 and 0.2, applying the same loss function (Average Relative Error), and setting the target image as a calculated standard noise map of the input images.

4. Dataset

The dataset comprises 10 subjects from the Grand Challenge Public Dataset. Each of those 10 subjects has 3 images (Figure 1). 8 of the 10 subjects were used for training, 1 subject for validation, and 1 subject for testing.

The images are described as follows:

a. CT signal - NRRD format, HU units
b. CT Noise - NRRD format, HU units
c. STD Noise Map - Integer representing Round (100 x STD), i.e. 2 significant figures. NRRD format.

5. Evaluation Metrics

To assess the performance of the model, both quantitative metrics and visual methods were used. The primary quantitative metric, Average Percentage Error (APE), measures the relative error between the predicted images and the ground truth, focusing on the mean prediction accuracy. Additionally, the model's effectiveness during testing is quantified through the observed testing loss.

$$Error = \left| \frac{Prediction - Ground\ Truth}{Ground\ Truth} \right|$$

(a)

$$APE = \frac{1}{n} \sum_{i=1}^{n} Error$$

(b)

Equation 3: (a) Relative Error, (b) Average Percentage Error

Visually, the model's performance is evaluated by examining the training-validation loss curves, which provide insights into the learning over iterations. Furthermore, to visually ascertain the model's predictive accuracy, the patches from the final predicted test images are reassembled into their original configuration. This reconstruction allows for a direct comparison with the original images, facilitating a deeper understanding of the model's practical effectiveness. In short, the success of the model is evaluated by two numeric metrics, namely, APE and Test Loss, and two modes of visual inspection via training-validation loss plots and predicted image patch reconstruction.

6. Results

One of the initial experiments conducted involved applying transformations to the dataset prior to training. Contrary to expectations, these transformations, particularly intensity normalization, adversely affected both training and validation performance, as depicted in Figure 6. Consequently, subsequent experiments were performed without any transformations.

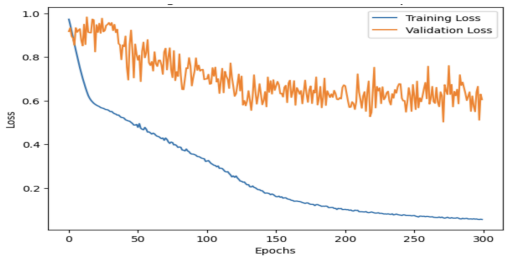
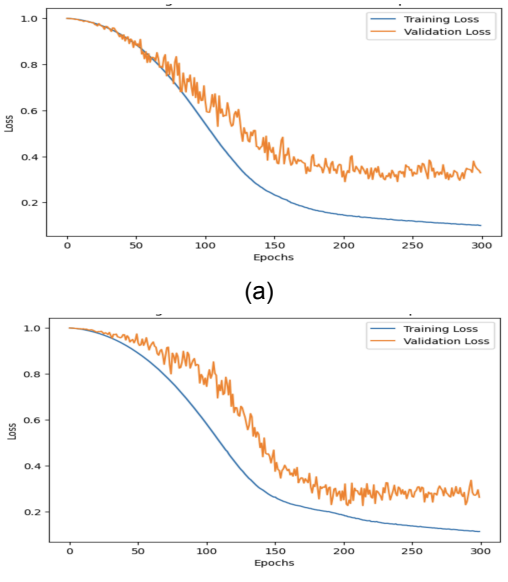


Figure 6: UNet validation showing consistently high loss with augmentation

For both the UNet and RatUNet models, the effects of incorporating dropout during the training process were explored. The models were trained under conditions both with and without dropout to evaluate their impact on model robustness and generalization.

The optimization of the models was carried out using the Adam optimizer. Specifically, a learning rate of 0.00001 was set for the RatUNet, while the UNet was optimized with a learning rate of 0.0001. To ensure the reliability of our results, experiments were carried out using three different initial seeds, averaging the outcomes to enhance the robustness and accuracy of the noise map predictions.

6.1 UNet Results



(a)

(b)

Figure 7: (a)UNet with a dropout rate of 0.1, (b) UNet without dropout

Table 1: UNet - Test Loss and APE

UNet	Test Loss	APE (%)
With dropout 0.1	0.20	14.90
Without dropout	0.14	6.75

With the intention of reducing overfitting, dropout was implemented at a rate of 0.1. However, this strategy appeared to adversely affect the model's accuracy, as evidenced by elevated test loss and Average Percentage Error (APE) values detailed in Table 1.

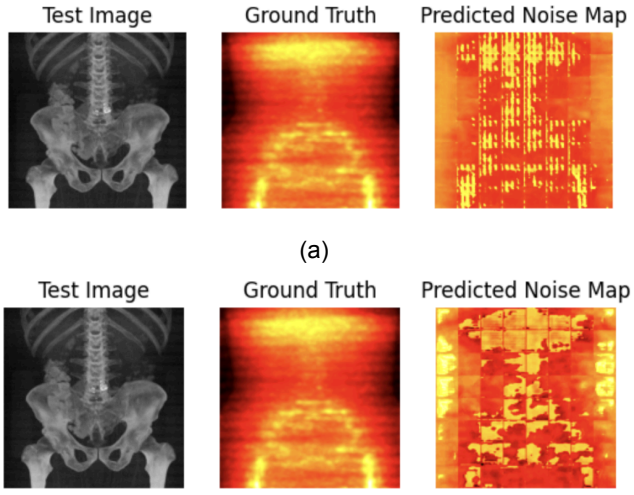


Figure 8: (a) Prediction with dropout, (b) Prediction without dropout

Figure 8 showcases the predicted images reassembled from patches after training. Figure 8(a) reveals that training with dropout resulted in predictions that failed to fully estimate the noise map, which is numerically supported by the higher test loss and APE. In contrast, Figure 8 (b) shows the prediction made by the model without dropout. This method showed lower test loss and APE and the predicted image was also visually more similar to the expected ground truth.

6.2 RatUNet Results

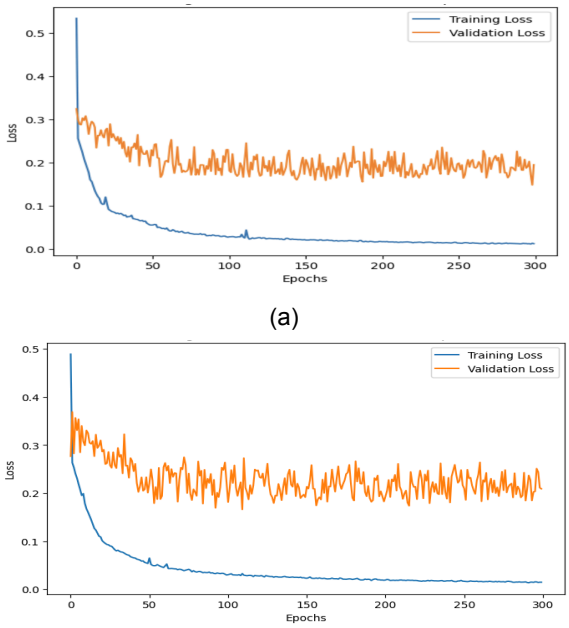


Figure 9: (a)RatUNet with a dropout rate of 0.2, (b) RatUNet without dropout

Table 2: RatUNet - Test Loss and APE

RatUNet	Test Loss	APE (%)
With dropout 0.2	0.16	13.75
Without dropout	0.16	12.72

Note: Dropout rates of 0.2 were tested for RatUNet because using dropout rates of 0.1 did not have a significant influence on results.

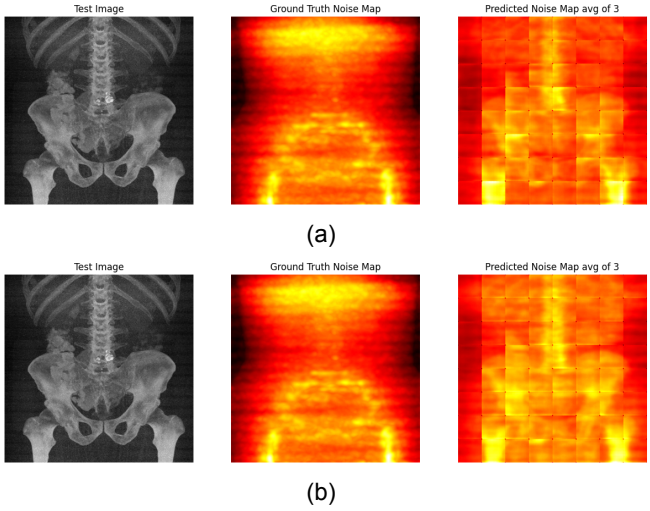


Figure 10: (a) Prediction with dropout, (b) Prediction without dropout

Figure 10 presents the average predicted images from the RatUNet model, following patch reconfiguration. According to the results in Table 2, altering the dropout

rate did not enhance performance; furthermore, RatUNet's predictions did not surpass those achieved by the UNet model without dropout. On average, the best prediction from the RatUNet introduced approximately 13% error relative to the ground truth and tended to overestimate the noise map. Visually, these predictions performed poorly in the upper section of the noise map.

7. Conclusions

Initial experiments demonstrated that data transformations such as intensity normalization, rotations, and reflections negatively impacted model performance. Models trained without these augmentations exhibited reduced loss and improved predictions, corroborated by both visual and numerical validation methods. Notably, the UNet architecture outperformed RatUNet in terms of training stability and smoothness. Since RatUNet is a more complex architecture, mainly designed for denoising, the initial assumption was that a network capable of denoising could potentially be capable of quantifying the noise which would eventually help in predicting STD noise maps. This initial assumption turned out to be somewhat relevant. While the RatUNet did a decent job of predicting the noise maps, it tended to overestimate the noise, potentially due to the complexity of its attention mechanism.

Both UNet and RatUNet showed minimal influence with the inclusion or exclusion of dropouts in their configurations, with numeric evaluations suggesting that models trained without dropouts achieved comparatively better results. This was partially confirmed through visual assessments. Overall, UNet demonstrated better generalization capabilities than RatUNet.

A significant challenge encountered in this project was the limited availability of data, which may have hindered the models' learning, particularly as data augmentations did not enhance the models' robustness. Future experiments could explore the same models with more training data, or combine datasets from different image reconstruction planes, and the next steps could potentially involve formatting the images and fine-tuning the model to make it train well with augmentations, in addition to increasing the training data. Besides, employing overlapping patches could facilitate smoother predictions. Consideration of pre-trained models as opposed to training from scratch may also yield different outcomes, potentially improving model effectiveness.

References

[1] N. R. Huber, J. Kim, S. Leng, C. H. McCollough, and L. Yu, "Deep Learning-Based Image Noise Quantification Framework for Computed Tomography," *J. Comput. Assist. Tomogr.*, vol. 47, no. 4, pp. 603-607, Jul.-Aug. 2023. doi: 10.1097/RCT.0000000000001469. Epub June 30, 2023. PMID: 37380148; PMCID: PMC10363183.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham: Springer, 2015, vol. 9351, pp. 234–241. doi: https://doi.org/10.1007/978-3-319-24574-4_28

[3] H. Zhang, Q. Lian, J. Zhao, Y. Wang, Y. Yang, and S. Feng, "RatUNet: Residual U-Net Based on Attention Mechanism for Image Denoising," *PeerJ Comput. Sci.*, vol. 8, e970, May 2022. doi: 10.7717/peerj-cs.970. PMID: 35634105; PMCID: PMC9138094.

[4] X. Shen, L. Lin, X. Xu, and S. Wu, "Effects of Patchwise Sampling Strategy to Three-Dimensional Convolutional Neural Network-Based Alzheimer's Disease Classification," *Brain Sci.*, vol. 13, no. 2, p. 254, Feb. 2023. doi: 10.3390/brainsci13020254. PMID: 36831797; PMCID: PMC9953929

[5] M. H. Alkinani and M. R. El-Sakka, "Patch-based models and algorithms for image denoising: A comparative review between patch-based images denoising methods for additive noise reduction," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, art. no. 58, 2017. doi: 10.1186/s13640-017-0203-4. Epub Aug 24, 2017. PMID: 32010201; PMCID: PMC6961526

[6] S.Ioffe and C.Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proc. 32nd Int. Conf. on Machine Learning - Vol. 37 (ICML'15)*, 2015, pp. 448-456. arXiv preprint arXiv:1502.03167, revised Mar. 2015. doi: 10.48550/arXiv.1502.03167

Appendix A. Detailed Roles

Table 3. Team member contributions

Name	Task	File names	No. Lines of Code
Avantika Kothandaraman	UNet - Training from scratch and validation, Evaluation, Present Slides, Paper Writing	All .py files in the UNet folder of the repository	397
Caiwei Zhang	RatUNet - Training from scratch and validation, Evaluation, Present Slides, Paper Writing	Files ratunet.py, train.py, and train_and_test.py in the RatUNet folder of the repository	375
Long Chen	Custom Dataset, Testing, Evaluation, Present Slides, Paper Graphics	README of repository	

Appendix B. Code GitHub Repository

Link: https://github.com/Maggiemajiii/STD-Noise-Maps-of-CT-Images.git