Journal of Mathematical Imaging and Vision (2025) 67:45
https://doi.org/10.1007/s10851-025-01260-y

Texture Matching GAN for CT Image Enhancement

Madhuri Nagare[1,2] · Gregery T. Buzzard[3] · Charles A. Bouman[1]

Received: 20 November 2024 / Accepted: 24 June 2025

Abstract

Deep neural networks (DNNs) are commonly used to denoise and sharpen X-ray computed tomography (CT) images with the goal of reducing patient X-ray dosage while maintaining reconstruction quality. However, naive application of DNN-based methods can result in image texture that is undesirable in clinical applications. Alternatively, generative adversarial network (GAN)-based methods can produce appropriate texture, but naive application of GANs can introduce inaccurate or even unreal image detail. In this paper, we propose a texture matching generative adversarial network (TMGAN) that enhances CT images while generating an image texture that can be matched to a target texture. We use parallel generators to separate anatomical features from the generated texture, which allows the GAN to be trained to match the desired texture without directly affecting the underlying CT image. We demonstrate that TMGAN generates enhanced image quality while also producing image texture that is desirable for clinical application.

Keywords Low-dose CT · Texture matching · Denoising · Sharpening · Generative adversarial network

1 Introduction

X-ray computed tomography (CT) is one of the most widely used 3D medical imaging modalities, with recent progress on reconstruction methods resulting in reduced noise and artifacts while improving resolution and quality [1]. In particular, noise reduction and image sharpening methods can be used to reduce X-ray dosage while maintaining image quality. However, the true measure of quality for a medical CT reconstruction method is its ability to improve diagnostic accuracy.

Radiologists regard the texture of CT images after enhancement as critically important for diagnosis [2]. In fact, high-quality texture provides important visual cues in decision making for radiologists [3]. Since radiologists are typically familiar with the noise texture of filtered back-projection (FBP) [4], this texture is often described as desirable [5, 6]. Quantitatively, [7] found that reducing noise while maintaining a texture like that of FBP led to better lesion detection than noise reduction that changed the texture.

Approaches to control CT image texture include [8, 9], which synthesize high-quality texture by matching the statistics of generated texture to a target. However, employing this method while denoising or sharpening a CT image requires separation of true anatomy from texture. This separation can be done using morphological component analysis (MCA) [10] as in [11, 12]. However, MCA requires dictionary learning for both the object and texture components, which is computationally expensive.

From [13], adaptive methods to preserve textural information include patch-based approaches using spatial similarity [14–16], which employ parameters to control image smoothness. Alternatively, [17] used a tuned a prior distribution in an iterative reconstruction [18] to produce desirable texture. However, these methods tend to be computationally expensive.

Deep neural networks (DNNs) are currently among the most popular methods for CT image enhancement [19, 20]. CNN denoisers typically require training with a loss function, and the mean squared error (MSE) loss function is

Charles A. Bouman was partially supported by the Showalter Trust.

✉ Madhuri Nagare
madhurinagare@gmail.com

Gregery T. Buzzard
buzzard@purdue.edu

Charles A. Bouman
bouman@purdue.edu

[1] School of ECE, Purdue University, West Lafayette, IN, USA

[2] Present Address: Apple, Inc., Cupertino, CA, USA

[3] Department of Mathematics, Purdue University, West Lafayette, IN, USA

Published online: 22 July 2025

commonly used [21] because it approximately maximizes the peak signal-to-noise ratio (PSNR). However, it is also known that the MSE loss function tends to produce images that are overly smooth and lack texture [22].

Deep learning image reconstruction (DLIR) is a proprietary algorithm of GE HealthCare that is marketed under the tradename *TrueFidelity* [23]. While the details of DLIR are not publicly available, it is based on direct neural network image domain denoising. DLIR produces CT images that preserve texture similar to high dose filtered back-projection (FBP) reconstruction and are shown to have a noise power spectrum similar to that of the FBP reconstruction in [23]. However, this noise spectrum is measured only in uniform regions of the reconstruction.

Other DNN approaches include [24, 25], which used a novel loss function to preserve texture and detail while denoising. The algorithm in [26] used a method to create training pairs by adding appropriately scaled noise to both input and ground truth, so that a DNN can be trained to preserve texture. Liu et al. propose incorporating a loss term that compares noise power spectra to enhance detail in reconstructions, thereby improving texture [27]. However, none of these approaches provide the ability to specify a target texture distribution.

Another approach to improving texture using DNNs is to use a generative adversarial network (GAN) [22], with the generator output (i.e., the denoised or sharpened image) as an input to the discriminator [28–31]. While these approaches can produce more realistic texture, they do not allow optimization to produce a particular desirable target texture. Also, in GAN-based methods, since the underlying CT image is not separated from the texture, the discriminator could encourage the addition of inaccurate or even unreal image detail known as hallucinations [32]. Xian et al. [33] used a conditional GAN to produce a target texture in natural images when an image sketch is provided.

Diffusion models have shown promise in enhancing CT images by iteratively denoising or reconstructing high-quality scans from noisy or undersampled data, demonstrating superior performance in preserving anatomical detail and suppressing artifacts [34, 35]. However, diffusion methods are computationally expensive to apply for inference, so they are less desirable for application in enhancing large medical image volumes [36]. Moreover, it is not clear how to apply this approach to produce desired target textures.

In this paper, we propose the texture matching GAN (TMGAN), which denoises and/or sharpens CT images while simultaneously matching the generated texture to a distribution of target textures. The methods of TMGAN build on our earlier research presented in [24]. A novel aspect of TMGAN is that it separates the texture from the underlying clean CT image by adding two independent noise samples to the same ground truth image and processing them with a Siamese network [37] (generator) to produce two conditionally independent estimates. We take the difference of these two estimates to separate the texture component from the underlying clean CT image. This allows the GAN to be trained without the risk of generating false image detail or hallucinations.

Our main contribution is a TMGAN architecture that:

- Denoises or sharpens a CT image while generating image texture that matches a desired target texture;
- Separates noise texture from the underlying clean CT image by subtracting two conditionally independent estimates with the same ground truth;
- Uses a novel bias reduction method to reduce bias in the estimated image;

We demonstrate the effectiveness of the TMGAN approach on simulated and experimentally measured CT data and show both quantitatively and qualitatively that TMGAN yields better texture quality than existing approaches.

2 Problem Formulation

Let Y be an observed image (in the reconstruction or image domain) from which we aim to recover the true image X. We use the following forward model for training our algorithm,

$$Y = G(X) + W, \tag{1}$$

where W is noise and $G(\cdot)$ models other possible deformations in Y; for example, in sharpening applications, $G(\cdot)$ is a blurring function. Note that W is not assumed to be independent of X, and in some practical cases the noise amplitude can depend on the signal strength in CT reconstruction. We then seek to estimate X as $\hat{X} = h(Y)$.

To model texture, we define the image estimate $\hat{X}$ as a sum of three components,

$$\hat{X} = X + B_X + \delta_X, \tag{2}$$

where we formally define

$$B_X = \mathbb{E}[\hat{X}|X] - X \tag{3}$$

$$\delta_X = \hat{X} - \mathbb{E}[\hat{X}|X]\,, \tag{4}$$

where B_X is the bias in the estimate and δ_X is the estimation noise, or texture, with $\mathbb{E}[\delta_X|X] = 0$ from (4). Note that if X is known, then B_X is deterministic, whereas δ_X is still a random variable and a function of X.

We highlight that δ_X is not the same as W. The estimation noise, δ_X, represents the noise texture in our estimate. Consequently, our goal will be to match the texture of δ_X to

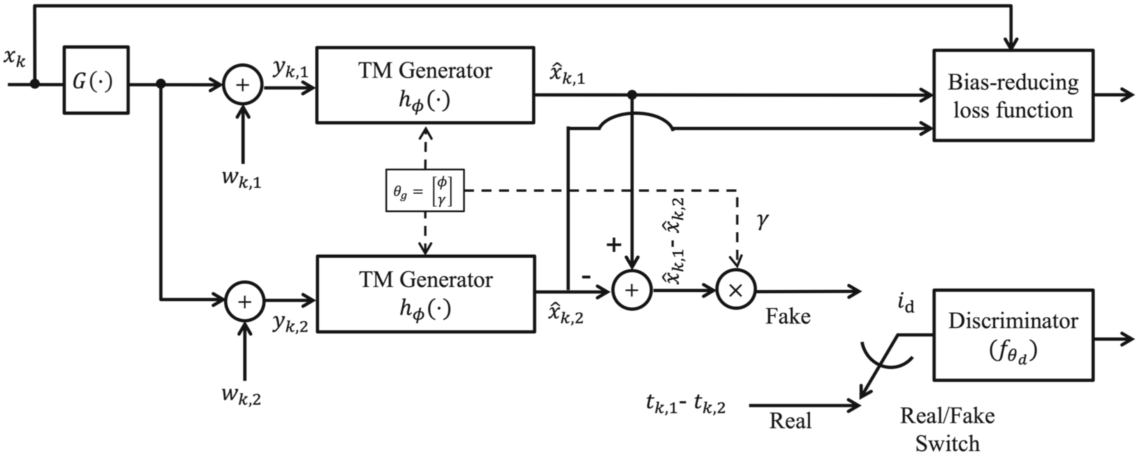

x_k
$G(\cdot)$
$y_{k,1}$
TM Generator $h_\phi(\cdot)$
$\hat{x}_{k,1}$
Bias-reducing loss function
$w_{k,1}$
$\theta_g = \begin{bmatrix}\phi\\ \gamma\end{bmatrix}$
γ
+
$\hat{x}_{k,1}$- $\hat{x}_{k,2}$
-
TM Generator $h_\phi(\cdot)$
$y_{k,2}$
$\hat{x}_{k,2}$
Fake
i_d
Discriminator (f_{θ_d})
$w_{k,2}$
$t_{k,1}$- $t_{k,2}$
Real
Real/Fake Switch

Fig. 1 Network architecture for Texture Matching GAN (TMGAN)

user-provided independent samples of the target noise texture, T. In other words, our goal is to train a conditional GAN with output $\hat{X} = h(Y)$ such that the resulting δ_X has the desired target distribution.

However, this is difficult because during the training process we never directly observe δ_X. Instead, we directly observe $X - \hat{X} \neq \delta_X$. Consequently, it is not immediately clear how to design a loss function for this task.

The key to training TMGAN is to add two noise instances W_1 and W_2 that are conditionally independent given a single ground truth X in (1). The resulting Y_1 and Y_2 are used to give estimated images $\hat{X}_1 = h(Y_1)$ and $\hat{X}_2 = h(Y_2)$, which are then conditionally independent given X. This implies that the resulting δ_{X_1} and δ_{X_2} are also conditionally independent given X.

From this we can see that

$$\begin{aligned} &\hat{X}_1 - \hat{X}_2 && (5)\\ &\quad = (X + B_X + \delta_{X_1}) - (X + B_X + \delta_{X_2}) \\ &\quad = \delta_{X_1} - \delta_{X_2}, && (6) \end{aligned}$$

where δ_{X_1} and δ_{X_2} are conditionally independent and identically distributed (i.i.d.) given X.

Based on this analysis, we will design the GAN so that

$$\hat{X}_1 - \hat{X}_2 \sim T_1 - T_2 \,, \tag{7}$$

where $\hat{X}_1$ and $\hat{X}_2$ are generated using conditionally independent noise samples and T_1 and T_2 are independent samples of the target texture, and we use $\sim$ to indicate the two differences have the same distribution.

Of course, if the texture has the desired distribution, then δ_{X_k} and T_k have the same distribution, and (7) must hold. However, the converse is not necessarily true. So, if (7) holds, it is not necessarily the case that δ_{X_k} and T_k have the same distribution, which is our goal. In other words, (7) is a necessary but not sufficient condition to ensure the desired texture distribution.

Theorem 1 establishes the sufficiency under the strong assumption of Gaussian differences, so that in this case, (7) ensures the desired texture distribution, which provides some theoretical motivation for our approach.

Theorem 1 *Let* $\delta_1 - \delta_2 \sim \mathcal{N}(0, 2\sigma^2)$ *and* δ_1, δ_2 *be real valued i.i.d. random variables with a distribution that is symmetric about* 0*. Then* $\delta_1 \sim \mathcal{N}(0, \sigma^2)$.

We note that in real scans, the actual distribution of image noise may not be well approximated by Gaussian, and the theorem assumptions do not hold. However, our experimental results will demonstrate that even under more general conditions, enforcing (7) leads to textures that match the desired target texture distribution.

3 Texture Matching GAN

3.1 TMGAN Architecture

Figure 1 shows the network architecture for TMGAN, where lower-case letters denote samples of the aforementioned random variables. The central component is the TM generator, $h_\phi(\cdot)$, a neural network parameterized by the vector ϕ, which uses noisy and distorted input, y, to estimate the original image as $\hat{x} = h_\phi(y)$. As described below, we optimize h_ϕ not only to minimize mean squared error (MSE) but also to produce a texture that statistically matches the provided training or target texture samples. Since the training texture samples may have a different amplitude than the estimation texture, we use a parameter, γ, to account for the potential difference in scaling. This γ parameter can be learned or set manually.

For the k^{th} ground truth image, we first apply the deformation $G(\cdot)$ (Gaussian blur for sharpening and the identity for denoising) and then generate two conditionally independent inputs $y_{k,1}$ and $y_{k,2}$ by adding independent noise samples $w_{k,1}$ and $w_{k,2}$. From these two inputs, the TM generator produces two estimates, $\hat{x}_{k,1} = h_\phi(y_{k,1})$ and $\hat{x}_{k,2} = h_\phi(y_{k,2})$. Following (5), we take the scaled difference $\gamma(\hat{x}_{k,1} - \hat{x}_{k,2})$ to get a sample of the fake texture difference, $\gamma(\delta_{x1} - \delta_{x2})$. We also generate samples of the real texture difference $t_{k,1} - t_{k,2}$ using sample images of the target texture by taking pixel-wise difference.

3.2 TMGAN Training

To train the TM generator, we need to promote accurate image estimation and match generated texture differences to target texture differences. To match texture differences, we use a GAN framework [38] in which the discriminator network is trained to differentiate scaled fake texture differences from real texture differences. We model the discriminator as a function $f_{\theta_d}(\cdot) \in (0, 1)$, with parameters θ_d, and we interpret f_{θ_d} as the probability that a texture difference sample is real.

Using this notation, the discriminator is trained by minimizing the binary cross-entropy (BiCE) [38] loss function with respect to θ_d:

$$d(\theta_g, \theta_d) = -\frac{1}{K}\sum_{k=1}^{K}\Big[\log\big(f_{\theta_d}(t_{k,1} - t_{k,2})\big) + \log\big(1 - f_{\theta_d}(\gamma(h_\phi(y_{k,1}) - h_\phi(y_{k,2})))\big)\Big], \quad (8)$$

where $\theta_g = [\phi, \gamma]$ and K is the number of training samples.

Unlike previous GAN-based methods for CT image enhancement, the TMGAN discriminator works only on the texture part, which avoids the risk of possible addition of fake detail, known as hallucinations, in the enhanced images.

Since our goal is to match ground truth while maximizing texture quality, we need two loss terms for the generator, one to promote desirable textures and a second to minimize MSE. The MSE term incorporates the bias-reducing loss function of [24], which demonstrated that the bias-reducing loss function yields better structural detail in denoised images than was obtained with the standard MSE loss.

Using this strategy, the TMGAN generator loss function is

$$g(\theta_g, \theta_d) = \frac{1}{K}\sum_{k=1}^{K}\Bigg[-\lambda \log f_{\theta_d}\big(\gamma(h_\phi(y_{k,1}) - h_\phi(y_{k,2}))\big) + \frac{1}{2\sigma^2}\Big(\|\hat{z}_{k,1} - x_k\|^2 + \|\hat{z}_{k,2} - x_k\|^2\Big)\Bigg], \quad (9)$$

where

$$\hat{z}_{k,1} = \alpha\hat{x}_{k,1} + (1-\alpha)\hat{x}_{k,2}$$
$$\hat{z}_{k,2} = (1-\alpha)\hat{x}_{k,1} + \alpha\hat{x}_{k,2}$$

form the basis of the bias-reducing loss, λ weights texture loss versus fit to data, and σ is roughly the standard deviation in estimating x_k. We set $\alpha = 0.5$ for maximum bias reduction for denoising [24] and $\alpha = 1.0$ to reduce aliasing in the sharpening case. Note that each of the two branches of the TM Generator network share the same parameters, so this can be treated as a Siamese network [37] for training.

Algorithm 1 shows the pseudocode to train the TM generator. To avoid common instability and nonconvergence issues encountered in training GANs [39], we use a training procedure to reduce mode collapse and vanishing gradients. An optimal discriminator can help avoid mode collapse [40], so we use multiple discriminator updates between generator updates. However, an optimal discriminator might not provide enough gradient to the generator to make progress, so to avoid vanishing gradients, we update discriminator weights only if the loss is greater than a threshold T_d. For inference, we employ the trained $h_\phi(\cdot)$ only.

Algorithm 1: Training Pseudocode for TMGAN

```
Input: $N$ : total number of generator updates,
       $T_d$ : threshold for the discriminator loss,
       $N_d$ : maximum discriminator updates per generator update

$\theta_g, \theta_d \leftarrow$ Initialize network parameters.
for $n \leftarrow 0$ to $N$ do
    $x_k, w_{k,1}, w_{k,2}, t_{k,1}, t_{k,2} \leftarrow$ randomly select clean, noise,
    target texture images
    $y_{k,1} \leftarrow G(x_k) + w_{k,1}$
    $y_{k,2} \leftarrow G(x_k) + w_{k,1}$
    $n_d \leftarrow 0$
    while $d(\theta_g, \theta_d) > T_d$ AND $n_d < N_d$ do
        $\theta_d \leftarrow$ Update $\theta_d$ with one iteration of Adam optimizer to
        minimize $d(\theta_g, \theta_d)$ in (8)
        $n_d \leftarrow n_d + 1$
    end
    $\theta_g \leftarrow$ Update $\theta_g = [\phi, \gamma]$ with one iteration of Adam
    optimizer to minimize $g(\theta_g, \theta_d)$ in (9)
end
```

3.3 Blending

To provide fine-grained control over the amount of texture, we use a blending or averaging scheme between TMGAN and standard image estimation. When $\lambda = 0$ in (9), TMGAN reduces to the bias-reducing network of [24], which we call BR-α. This gives an accurate image estimate but without the target texture. The blended estimate is then

$$\hat{x}^{(\text{TMGAN-blended})} = \eta h_\phi(y) + (1-\eta) h_{BR-\alpha}(y), \quad (10)$$

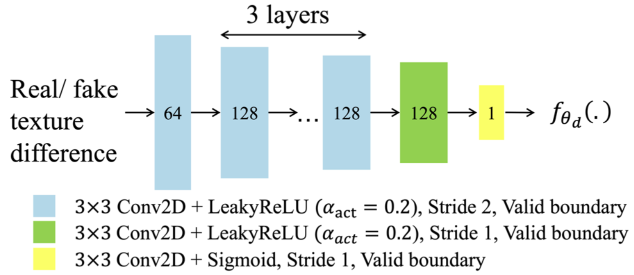

3 layers
Real/ fake
texture
difference
64
128
128
128
1
$f_{\theta_d}(.)$
3×3 Conv2D + LeakyReLU ($\alpha_{\text{act}} = 0.2$), Stride 2, Valid boundary
3×3 Conv2D + LeakyReLU ($\alpha_{act} = 0.2$), Stride 1, Valid boundary
3×3 Conv2D + Sigmoid, Stride 1, Valid boundary

Fig. 2 Discriminator architecture for TMGAN. The network is fully convolutional with a total of 6 layers

where y is the noisy input, η is the blending ratio, and $h_{BR-\alpha}(y)$ has no texture matching term.

When $\eta = 1.0$, the blended result is pure TMGAN, and when $\eta = 0.0$ it is pure image estimation. However, for intermediate values of η, it blends these two results, with larger values of η resulting in more texture, and smaller values resulting in reduced noise and more image detail.

4 Methods

4.1 TMGAN Implementation

We train TMGAN separately for the two applications of denoising and sharpening of noisy CT images. While the network architecture remains the same for both applications, the training data and test data are different.

For the generator architecture, we used a CNN adopted from [41] with a single input channel and 17 convolution layers. We modified the discriminator architecture in [42] to approximately match its capacity (the number of trainable parameters) to the generator. Figure 2 shows the discriminator architecture consisting of a series of 2D convolutional layers.

Table 1 lists the hyperparameters used in the TMGAN algorithm along with their significance, ranges and settings used in the experiments. The settings for λ and σ varied by experiment and are listed with the corresponding results. The values for all the parameters were selected empirically.

4.2 Datasets

All the scans used in training and evaluating TMGAN were acquired using a GE Revolution CT scanner (GE HealthCare, WI, USA).[1] The scans are reconstructed to a slice thickness of 0.625 mm and dimension 512 × 512. The standard (stnd) reconstruction kernel option available on the scanner and 40 cm DFOV (Display-Field-of-View) are used unless specified otherwise. The scans are reconstructed in HU units, and we added an offset of 1000 to all the images while training and testing so that air is 0.

4.2.1 Training and Validation Data

Ground truth images for denoiser training were generated by reconstructing 10 clinical scans with X-ray tube voltage and current varying from scan to scan in the range of 100-120 kVp and 445-1080 mA, respectively. The scans were reconstructed using the GE's TrueFidelity DLIR option [23].

Four ground truth images for sharpener training were obtained by averaging repeated scans of two distinct head phantoms in order to reduce noise. Each scan was acquired with a small focal spot size at 120 kVp/ 320 mA and reconstructed with a bone+ kernel [26] to a DFOV of 15 cm.

Noise and texture samples for training were generated by removing the mean from scans of 6 water phantoms, obtained with a tube voltage of 120 kVp, current of 350 to 380 mA.

For denoising, the deformation operator, $G(\cdot)$, was simply the identity operator. This was followed by the addition of two independent noise samples from the water phantoms to generate the two conditionally independent noisy input image samples, $y_{k,1}$ and $y_{k,2}$. For sharpening, the deformation operator, $G(\cdot)$, was the application of a Gaussian filter of standard deviation 0.244 mm, 0.244 mm, 0.344 mm in x, y, and z directions, respectively. This was again followed by the same process to form $y_{k,1}$ and $y_{k,2}$.

In both tasks, the GAN training and validation data was produced by breaking slices into 128 × 128 × 1 patches, with the patches randomly partitioned as 97% for training and 3% for validation. All DNN trainings were performed using the Adam optimizer [43] with a learning rate of 3×10^{-5} for the generator and 3×10^{-6} for the discriminator and a mini-batch size of 32. The network was implemented in TensorFlow [44] and trained with an NVIDIA Tesla V100 GPU.

4.2.2 Test Data and Evaluation Metrics

For quantitative evaluation, we generated realistic synthesized clean images by reconstructing nine clinical scans with the GE's TrueFidelity DLIR option [23]. The X-ray tube voltage and current varied from scan to scan in the range of 80–120 kVp and 55-375 mA, respectively. We also scanned a water phantom with a tube voltage of 120 kVp and current 350 mA. The clean volumes were used as ground truth, and noise from the water phantom was added to simulate the scanner noise. None of these scans were used in training. Using the ground truth, we computed the peak signal-to-noise ratio (PSNR) and SSIM (structural similarity) [45] metrics. We also show NPS (noise power spectrum) results computed using the methods of [46].

Table 2 lists the clinical exams used to test the algorithms. None of these exams were used in training. For each exam, the

[1] We thank GE HealthCare for collecting the datasets.

Table 1 Summary of parameters for TMGAN

Parameter and range	Description	Values used	
		Denoise	Sharp
$\alpha \in [0.5, 1.0]$	$\alpha \downarrow$ increases detail. See (9)	0.5	1.0
$\lambda \geq 0$	$\lambda \uparrow$ improves texture match See (9)	See results	
$\sigma \geq 0$	$\sigma \downarrow$ reduces squared error See (9)	See results	
$\eta \in [0.0, 1.0]$	Frac. of TMGAN in blended image (10)	0.3	1.0
$T_d > 0$	Threshold for Discr. update See Alg. 1	0.2	0.2
$N_d\ (\geq 1)$	Num. of Discr. updates per Gen update; See Alg. 1	1	5
$N \gg 1$	Total Gen. updates. See Alg. 1	$\approx 58K$	$\approx 228K$

Table 2 Clinical test exams

Exam name	Scanned object	Focal spot size	Dosage (kVp/mA)	DFOV (cm)
Exam 1	Body	XL	80/375	31.1
Exam 2	Body	Small	100/220	49.2
Exam 3	Body	Small	120/110	35.0
Exam 4	Body	Small	120/350	39.4
Exam 5	Head	XL	120/530	15

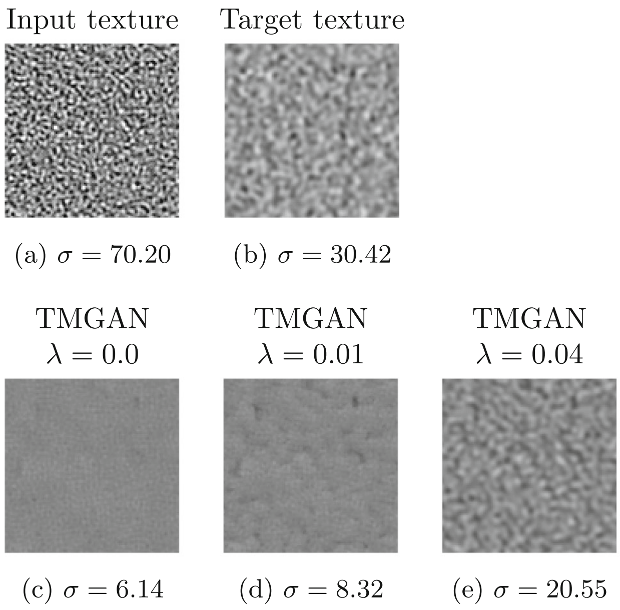
table lists the scan content along with various scan parameters. Exams 1-4 were used for the denoising experiment. Exam 5 was captured with an extra large (XL) focal spot size, which produced blurred features in the captured image; this exam was used for the sharpening experiment. Since these are clinical exams, no ground truth is available, so we provide qualitative evaluation only.

For denoising, TMGAN was compared to the following alternatives:

- **MSE denoiser**: Denoiser trained only with MSE loss
- **BR-**0.5: Denoiser using bias-reducing loss function with $\alpha = 0.5$ [24]
- **WGAN-VGG**: Method in [28] as used in [47]

For sharpening, TMGAN was compared to the following alternatives:

- **MSE sharpener**: Sharpener trained with MSE loss
- **NPSF**$_1$: NPSF sharpener [26] tuned to maintain the same level of noise as input



(a) $\sigma = 70.20$ (b) $\sigma = 30.42$

(c) $\sigma = 6.14$ (d) $\sigma = 8.32$ (e) $\sigma = 20.55$

Fig. 3 Comparison of TMGAN generated textures with standard deviations, σ, in HU with varying λ, the generator loss weight for texture matching (9). **a** Input to TMGAN (water phantom with bone+ filter recon), **b** target texture (water phantom with standard filter recon), result of TMGAN with **c** $\lambda = 0$, **d** $\lambda = 0.01$, and **e** $\lambda = 0.04$. Notice that texture in results becomes more similar to the target texture as λ increases

- **NPSF**$_2$: NPSF sharpener [26] tuned to achieve the same level of noise as TMGAN.

For a fair comparison, all denoisers and sharpeners, except WGAN-VGG, used the TM generator architecture.

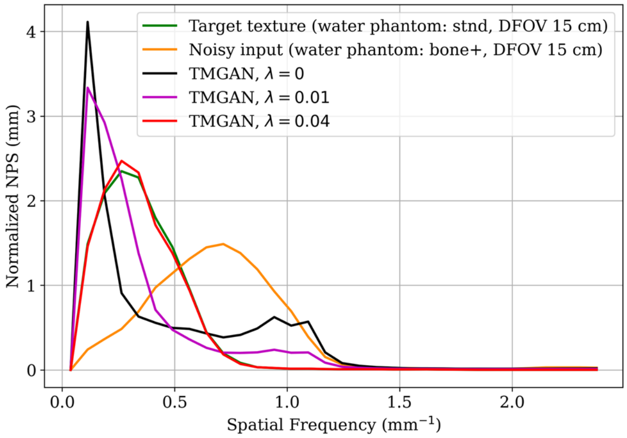


Fig. 4 NPS plots for texture images in Fig. 3. Notice that as λ increases, the shape of the NPS plots becomes more similar to the NPS of the target texture

5 Results and Discussion

5.1 Quantitative Evaluation

Figure 3 illustrates the textures generated by TMGAN with an input image of the test water phantom using the bone+ kernel and a target texture of a water phantom using the standard kernel. Both phantoms are reconstructed with a DFOV of 15 cm. Figure 3c–e shows the output of TMGAN with λ values of 0.0, 0.01, and 0.04, respectively, while σ was set to 50 HU. All images use a [-175, 175] HU window and a 5.86 cm FOV.

Figure 4 shows the corresponding NPS plots for all five images.

Consistent with the role of λ in controlling the importance of texture mapping in the generator loss function, as λ increases, TMGAN generates an output texture that matches more closely to the target texture both qualitatively and in terms of NPS plots. Moreover, for $\lambda = 0$, the NPS is skewed toward the lower frequencies, which is known to cause an overly smooth or “cartoony” texture in CT images [48].

Figure 3 also shows the standard deviation of the noise for each texture shown in Fig. 3. Again consistent with the role of λ, we see that the noise standard deviation increases as λ increases. This is expected since $\lambda = 0$ corresponds to minimizing the bias-reduced MSE only, while increasing λ promotes additional texture, which appears as increased noise energy. However, even with $\lambda = 0.04$ the TMGAN noise standard deviation is much less than the input noise standard deviation, indicating that TMGAN is able to simultaneously reduce noise and match texture.

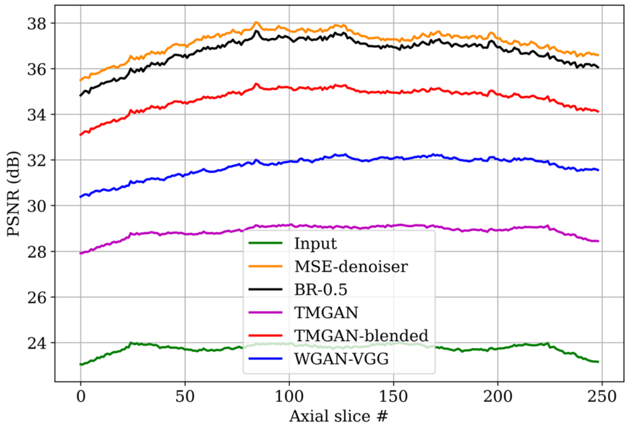


Fig. 5 Comparison of PSNR for axial slices of a synthetic exam. TMGAN trained with $\lambda = 0.4, \sigma = 7.8$ HU. TMGAN-blended preserves texture with a small reduction in PSNR

Table 3 PSNR and SSIM for nine synthetic test exams (mean ± 95 % Confidence Interval)

Method	PSNR	SSIM
Input	23.74 ± 0.0065	0.70 ± 0.0457
MSE denoiser	37.20 ± 0.503	0.81 ± 0.0588
BR-0.5	36.85 ± 0.529	0.82 ± 0.0523
WGAN-VGG	30.75 ± 0.366	0.77 ± 0.0588
TMGAN	28.98 ± 0.118	0.74 ± 0.0523
TMGAN-blended	34.87 ± 0.379	0.79 ± 0.0588

Figure 5 shows plots of the PSNR of axial slices for each of the denoising algorithms using one of the synthetic noisy exams as an input. The MSE denoiser generates the highest PSNR since it minimizes the MSE loss function. The BR-0.5 algorithm recovers more detail at the cost of a slight decrease in the PSNR. Since the GAN architectures of WGAN-VGG and TMGAN have a loss function that encourages texture recovery, they all have lower PSNR than the MSE denoiser and BR-0.5 methods. Finally, PSNR for TMGAN-blended is still lower than BR-0.5 but is higher than that of WGAN-VGG.

Table 3 lists the PSNR and SSIM values averaged over nine results obtained by inputting synthetic noisy exams to each algorithm. Notice that BR-0.5 has the best SSIM value. However, as we demonstrate next, the TMGAN-blended results produce much better texture with only a small decrease in the PSNR and SSIM.

Figure 6 compares the NPS of input and target textures along with the NPS of denoised results using various algorithms (whereas Fig. 4 varied λ). Both input and target textures were obtained by reconstructing the test water phantom with the standard kernel and a DFOV of 40 cm. The NPS for TMGAN most closely matches the target texture, while the NPS for TMGAN-blended has slightly increased low frequencies relative to TMGAN. More importantly, the NPS for TMGAN-blended matches the NPS of target texture at higher frequencies more closely than all other algorithms except TMGAN.

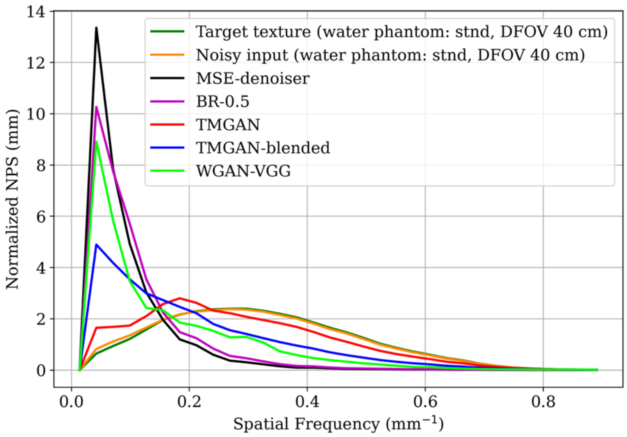


Fig. 6 NPS plots of water phantom reconstructions for denoising algorithms. TMGAN (trained with $\lambda = 0.4, \sigma = 7.8$ HU) produces closest match with the NPS of target texture. TMGAN-blended is second best with NPS slightly skewed toward the origin while preserving higher frequencies too

5.2 Qualitative evaluation with measured data

All exams in this section are measured low-dose scans, not simulated by adding noise to normal-dose scans.

Figures 7 and 8 show the results for two separate slices of Exam 1, a high-contrast clinical scan. Visually, results for the MSE denoiser have smooth texture and lack detail. In contrast, BR-0.5 has more detail than MSE denoiser, but with very nonuniform texture. The WGAN-VGG method recovers some texture and detail; however, the texture is not uniform.

From Figs. 7e and 8e, TMGAN produces uniform texture, but with increased noise variance and with some details obscured by the texture. Alternatively, TMGAN-blended achieves the desired uniform texture, along with reduced noise and more visible detail. More specifically, the arrows in Fig. 7f indicate the detail recovered by TMGAN-blended even while maintaining the uniform target texture of TMGAN.

To evaluate the computational load of TMGAN, Table 4 lists training and inference times for the networks compared in Fig. 7. TMGAN requires more training time per epoch compared to simple MSE denoiser or BR$-$0.5 networks

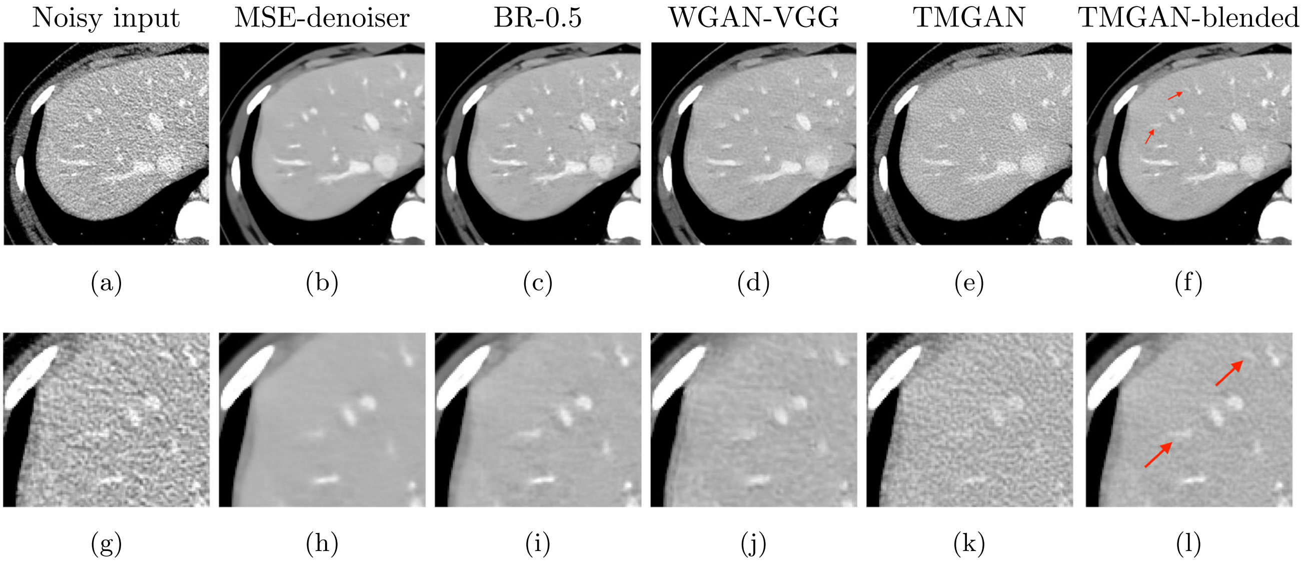


Fig. 7 Comparison of slice 1 of denoised results for Exam 1, a low-dose clinical scan. The second row shows zoomed ROI from the slice in the first row. TMGAN trained with $\lambda = 0.4, \sigma = 7.8$ HU. Display window is $[-125, 225]$ HU. BR$-$0.5 results maintain good detail, while TMGAN produces target texture which is more uniform and pleasing compared to other methods. With blending, we preserve detail (red arrows) from BR$-$0.5 and target texture in TMGAN (Color figure online)

Table 4 Training and inference times

Model	Epochs	Training time (Hrs.)	Inference time (Sec)
MSE denoiser	79	32.20	30
BR$-$0.5	76	29.71	30
WGAN-VGG	35	336	23
TMGAN	15	19.62	30

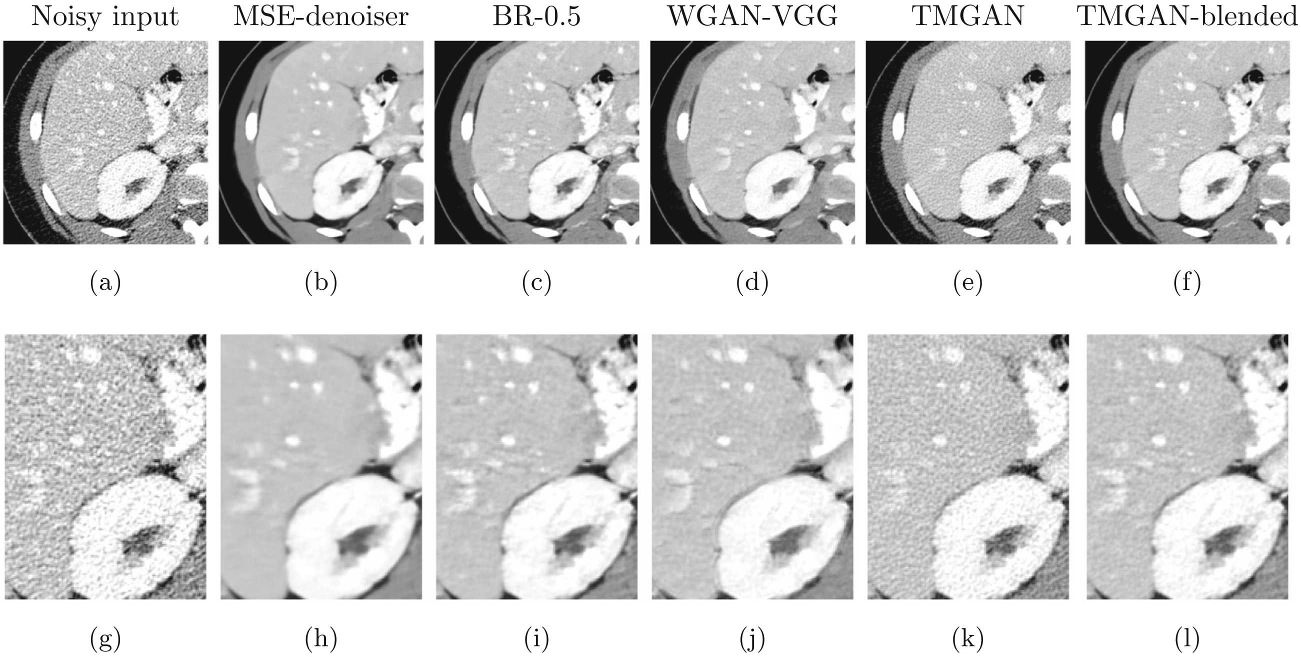


Fig. 8 Comparison of slice 2 of denoised results for Exam 1, a low-dose clinical scan. The first row shows full slice. The second row shows zoomed ROI. TMGAN trained with $\lambda = 0.4, \sigma = 7.8$ HU. Display window is [−125, 225] HU. TMGAN-blended produces target texture for a challenging input with very nonuniform texture

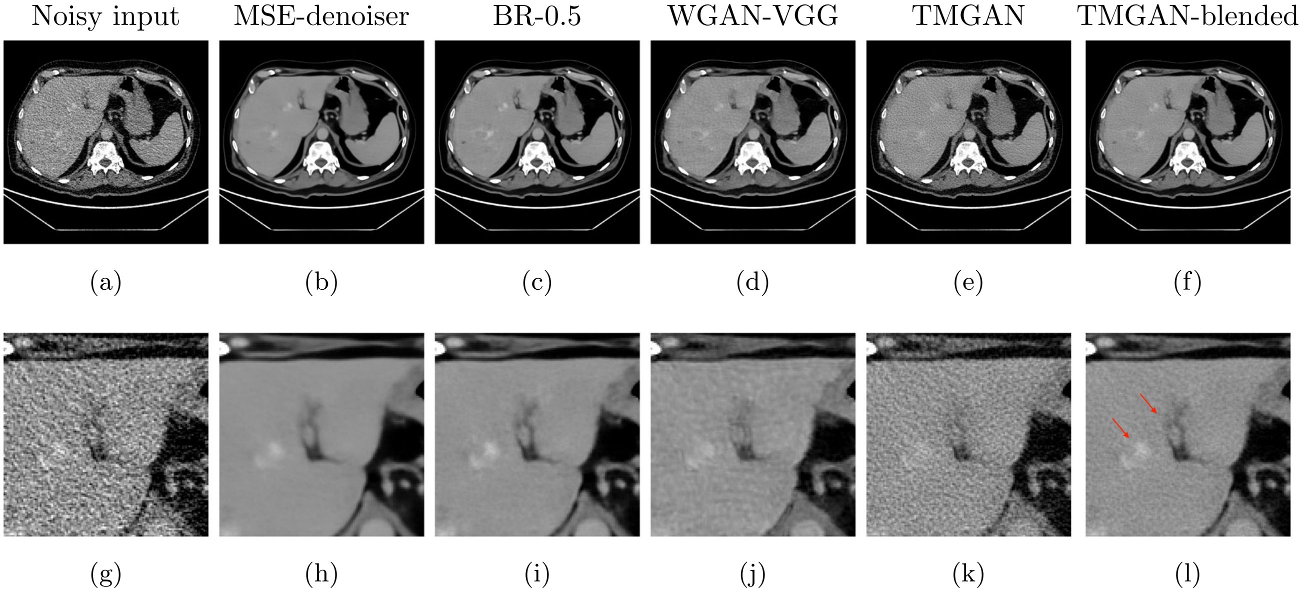


Fig. 9 Comparison of slice 1 of denoised results for Exam 2, a low-dose low-contrast clinical scan. The first row shows full slice. The second row shows zoomed ROI. TMGAN trained with $\lambda = 0.4, \sigma = 7.8$ HU. Display window is [−125, 225] HU. BR−0.5 results maintain good detail, while TMGAN produces target texture which is more uniform and pleasing compared to other methods. With blending, we preserve detail (red arrows) from BR−0.5 and target texture in TMGAN (Color figure online)

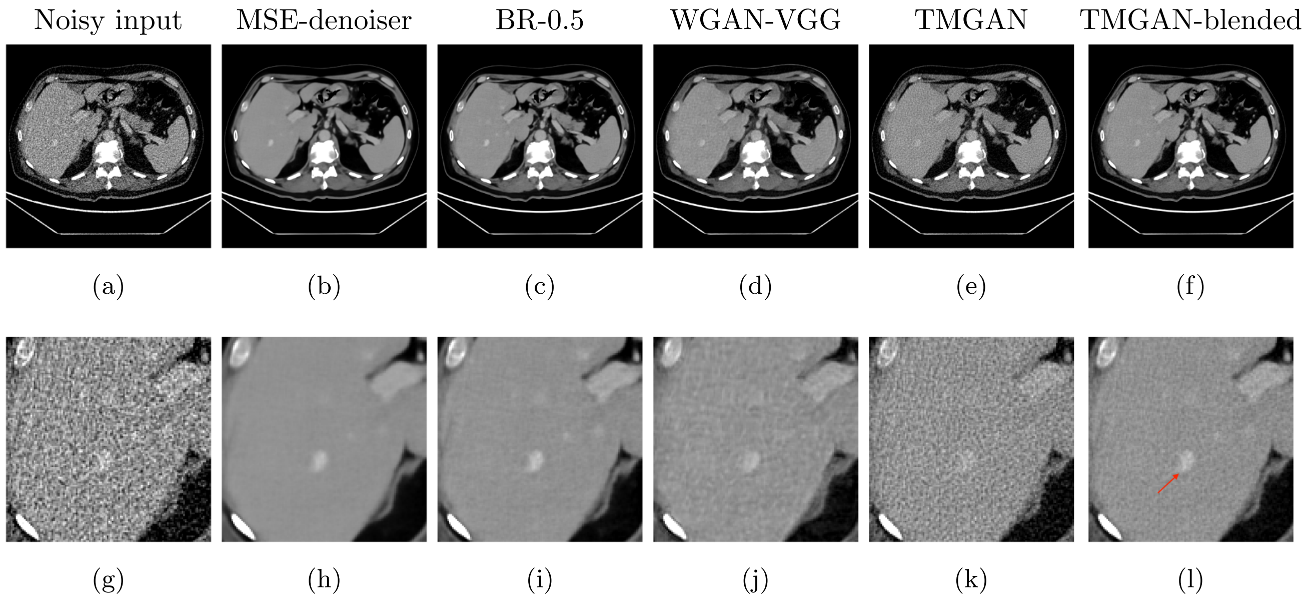


Fig. 10 Comparison of slice 2 of denoised results for Exam 2, a low-dose low-contrast clinical scan. The first row shows full slice. The second row shows zoomed ROI. TMGAN trained with $\lambda = 0.4$, $\sigma = 7.8$ HU. Display window is $[-125, 225]$ HU. BR$-$0.5 results maintain good detail, while TMGAN produces target texture which is more uniform and pleasing compared to other methods. With blending, we preserve detail (red arrow) from BR$-$0.5 and target texture in TMGAN

since TMGAN uses a more complex generator–discriminator architecture. However, we found that TMGAN gives good quality texture with fewer epochs for lower total wall time for training. More importantly, the inference time for TMGAN is still comparable to other networks and can be applied in real time, which is crucial for implementation in commercial CT scanners.

Figures 9 and 10 show results for Exam 2, a low-contrast clinical exam. Both the full slice and zoomed views show that TMGAN produces a uniform texture for this low-contrast exam, while WGAN-VGG produces a uniform but coarser texture. More importantly, the arrows in Figs. 9l and 10l show that low-contrast features are best detected using the TMGAN-blended results.

Figure 11 shows results for Exam 3 in the lungs. Note that the small air pockets in the lungs, shown by the yellow arrows in Fig. 11f, have diagnostic value but are not clearly visible in the MSE denoiser results. On the other hand, as seen from zoom-ins in the second row, WGAN-VGG, BR-0.5, TMGAN and TMGAN-blended recover them in the denoised images.

Figure 12 shows the results of Exam 4, which is a challenging exam due to the very low contrast. As seen in Fig. 12f, l, TMGAN-blended produces the target texture and recovers most of the detail seen in WGAN-VGG, which produces a less desirable texture.

Figure 13 shows TMGAN results for sharpening an image consisting of noise and aliasing artifacts. From Fig. 13, it is evident that the MSE sharpener results are over-smooth and contain artifacts. While the NPSF$_1$ has more detail and texture, it retains some aliasing artifacts. If we tune NPSF to have noise power the same as the TMGAN results to get NPSF$_2$, then there is partial noise reduction and sharpening, but the aliasing artifacts then look worse. In comparison, the TMGAN results are sharper than the input (green arrow) and have more uniform texture with a lower noise level than NPSF$_1$. The TMGAN results have reduced aliasing artifacts while remaining sharp, as indicated by the yellow arrow.

Figure 14 compares the NPS for a sharpened phantom scan with uniform areas using the algorithms discussed here. The NPS for the MSE sharpener is skewed toward low frequencies producing the over-smooth texture observed in Fig. 13. The NPS for NPSF and TMGAN results is very similar and matches with the target texture.

It is evident from the above results that even though TMGAN is a generative network, unreal details are not visible in resultant images. This is because the generative component in TMGAN is limited to the texture or noise part, and unreal details would get embedded in the noise and remain unnoticeable to human eyes.

Figure 15 demonstrates TMGAN's ability to generalize when targeted anatomy scans are not available during training. In this example, the input is a brain CT scan reconstructed with a bone+ kernel. TMGAN-Brain, trained on well-curated data matching the test distribution, yields the best results. In contrast, TMGAN-LiverLung, trained on reconstructed lung

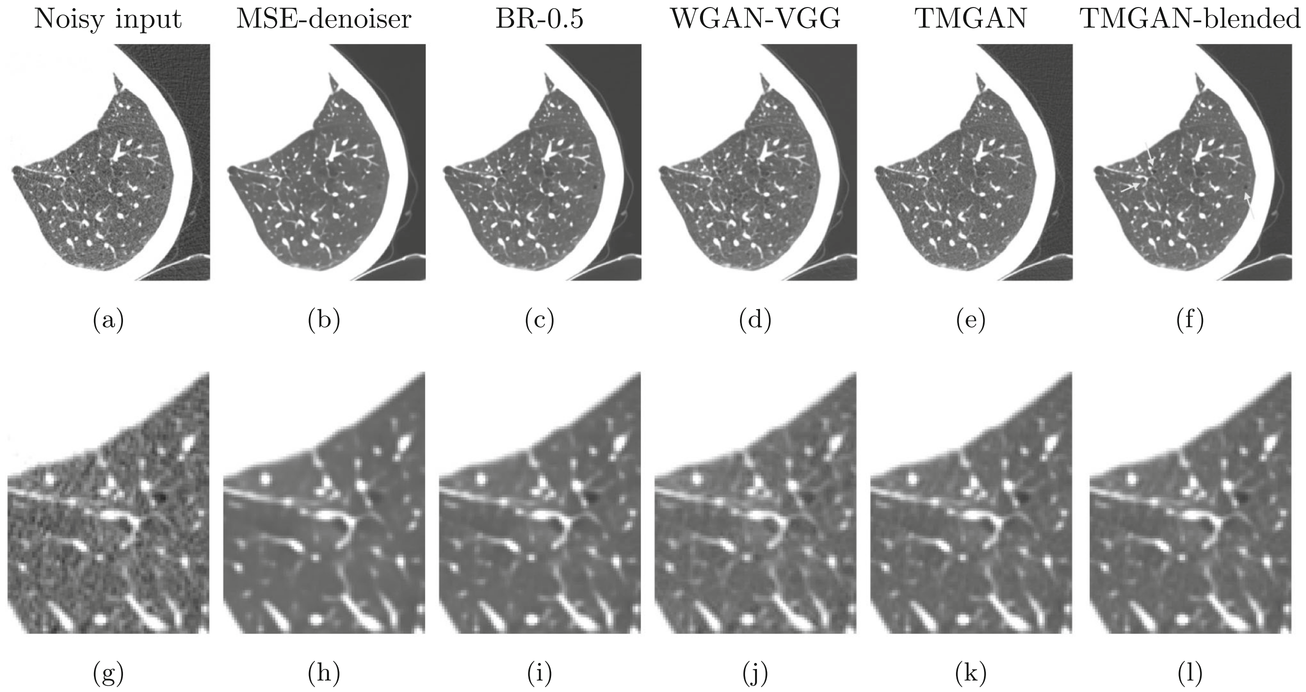


Fig. 11 Comparison of denoised results for Exam 3, a low-dose clinical lung scan. The first row shows full slice. The second row shows zoomed ROI. TMGAN trained with $\lambda = 0.4$, $\sigma = 7.8$ HU. Display window is $[-1200, -200]$ HU. BR-0.5 results maintain good detail, while TMGAN produces target texture which is more uniform and pleasing compared to other methods. With blending, we preserve detail (small air pockets indicated by yellow arrows) from BR-0.5 and nice texture in TMGAN

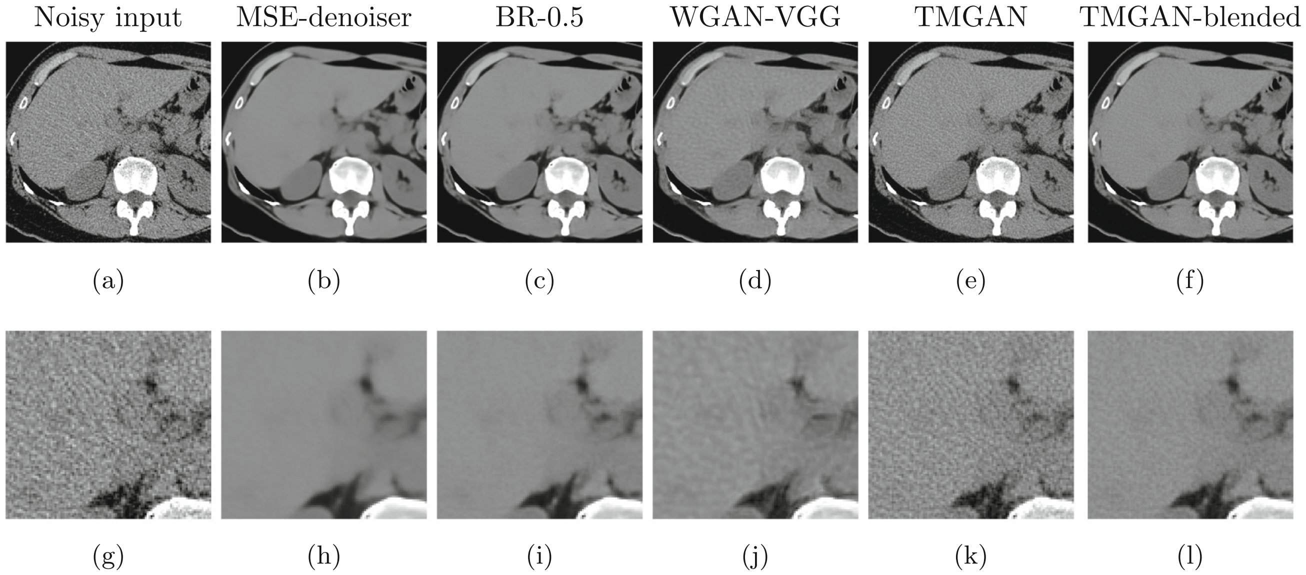


Fig. 12 Comparison of denoised results for Exam 4, a low-dose clinical scan. The first row shows full slice. The second row shows zoomed ROI. TMGAN trained with $\lambda = 0.4$, $\sigma = 7.8$ HU. Display window is $[-125, 225]$ HU. TMGAN-blended recovers good detail and a uniform target texture given a challenging input

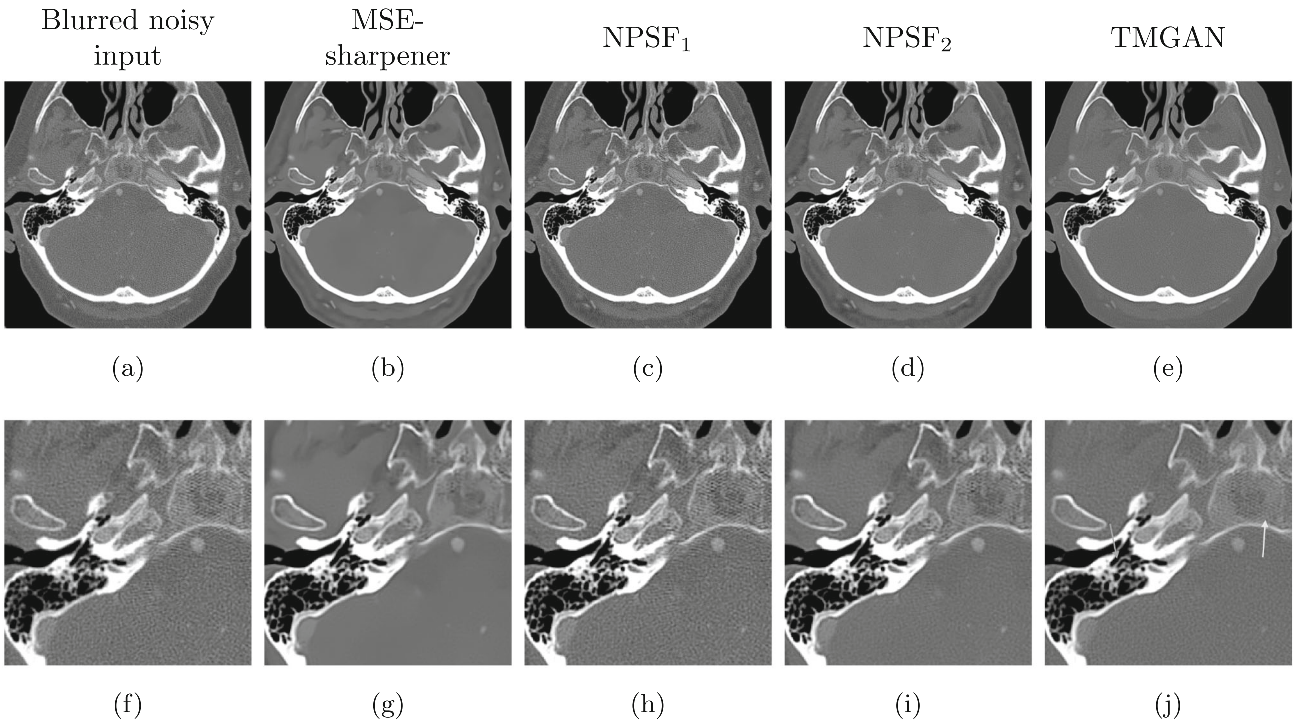


Fig. 13 Comparison of sharpening results for Exam 5, a low-dose and XL focal spot scan. The first row shows full slice. The second row shows zoomed ROI. TMGAN trained with $\lambda = 0.04, \sigma = 50$ HU. Display window is [−650, 1350] HU. TMGAN sharpens temporal bones (green arrow), while reducing aliasing artifacts (yellow arrow) (Color figure online)

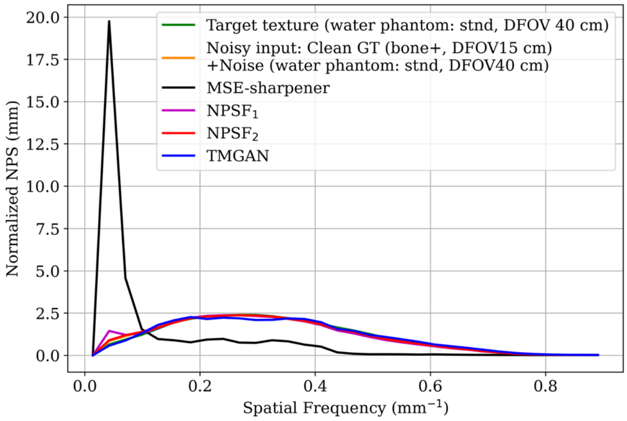


Fig. 14 Comparison of NPS for sharpening results. TMGAN and NPSF ($\beta > 0$) match the NPS of results to the target texture

and liver scans with a stnd kernel, still produces comparable results. The difference in sharpness arises because TMGAN-LiverLung was trained to denoise, while TMGAN-Brain was trained to deblur too. However, for clinical applications, we recommend training on data that matches the target distribution to ensure optimal accuracy.

6 Conclusion

We proposed a novel neural network, TMGAN, that denoises and/or sharpens CT images while simultaneously matching the texture of the resulting output to a target texture. We achieve this using a branched network with identical weights in each branch. Each branch processes the ground truth corrupted by noise, with the noise realization independent in the two branches. By taking the difference of the resulting outputs, our network separates texture from image with anatomical detail. By embedding this network in an adversarial training framework, we train to produce a desired texture layered on top of a clean image. The resulting output is an enhanced CT image that contains important physiological details and maintains a texture that is viewed as desirable by practicing radiologists. Our method reduces the risk of hallucination by separating the clean CT image containing anatomical features from texture and restricting generation to the texture part of the image. Furthermore, the bias–variance trade-off can be modulated as desired by using a simple blending method. Our experiments show that TMGAN removes streaking or aliasing artifacts and produces uniform texture while maintaining important detail.

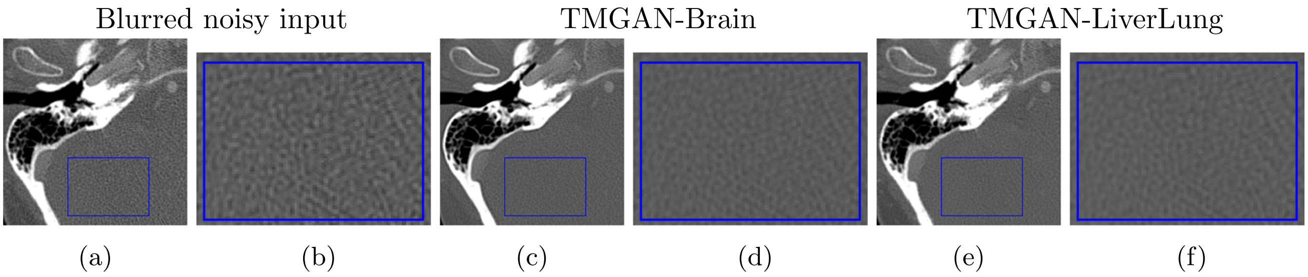


Fig. 15 Generalization capability of TMGAN when trained on one data set and applied on another. **a** Shows the input brain exam [bone+ kernel] and **b** shows the enlarged blue ROI, which has a noise std dev of 68.52 HU. **c** Shows the output of TMGAN trained with brain scans [bone+ kernel] and **d** shows the ROI, with a std dev of 37.92 HU. **e** shows the output of TMGAN trained with lung and liver scans [stnd kernel] and **f** shows the ROI with std dev of 36.57 HU. The display window is [−650, 1350] HU for all. TMGAN-LiverLung shows comparable performance to TMGAN-Brain despite mismatched training data

A Proof of Theorem 1

Theorem 2 *Let $\delta_1 - \delta_2 \sim \mathcal{N}(0, 2\sigma^2)$ and δ_1, δ_2 be real valued i.i.d. random variables with a distribution that is symmetric about* 0*. Then $\delta_1 \sim \mathcal{N}(0, \sigma^2)$.*

Proof Let $Z = \delta_1 - \delta_2$. Since Z, δ_1, and δ_2 are real valued, all of their characteristic functions exist. Since $Z \sim \mathcal{N}(0, 2\sigma^2)$, its characteristic function is given by

$$\phi_Z(t) = \mathbb{E}[e^{jtZ}] = e^{-\sigma^2 t^2} .$$

Since δ_1 and δ_2 are independent, we have that

$$\begin{aligned}\phi_Z(t) &= \mathbb{E}[e^{jt(\delta_1-\delta_2)}] \\ &= \mathbb{E}[e^{jt\delta_1} e^{-jt\delta_2}] \\ &= \mathbb{E}[e^{jt\delta_1}]\mathbb{E}[e^{-jt\delta_2}]\end{aligned}$$

Since δ_1 and δ_2 are symmetric about 0, we can remove the negative sign in the final expected value. Since δ_1 and δ_2 are i.i.d., the two expected values are the same, hence

$$\phi_Z(t) = \mathbb{E}[e^{jt\delta_1}]^2 .$$

Taking square roots yields

$$\mathbb{E}[e^{jt\delta_1}] = \pm e^{-\frac{1}{2}\sigma^2 t^2}$$

Since the characteristic function is always continuous, the choice of ± is independent of t. Since the left-hand side is 1 when $t = 0$, we see that the characteristic function of δ_1 is $e^{-\frac{1}{2}\sigma^2 t^2}$. By uniqueness of the characteristic function, this means that $\delta_1 \sim \mathcal{N}(0, \sigma^2)$, hence likewise for δ_2. □

Acknowledgements We thank Jonathan S. Maltz, Roman Melnyk, Brian Nett, and Ken D. Sauer for fruitful discussions and Karen Procknow for her valuable clinical feedback.

Author Contributions M.N. implemented the method and performed experiments. M.N. and C.A.B. wrote the main manuscript. All authors contributed to the theoretical results and edited the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Hsieh, J.: Computed Tomography: Principles, Design, Artifacts, and Recent Advances, 3^{rd} edn. SPIE Press, USA (2015)
2. Waite, S., et al.: Analysis of perceptual expertise in radiology-current knowledge and a new perspective. Front. Hum. Neurosci. **13**, 213 (2019)
3. Ghalati, M.K., Nunes, A., Ferreira, H., Serranho, P., Bernardes, R.: Texture analysis and its applications in biomedical imaging: a survey. IEEE Rev. Biomed. Eng. **15**, 222–246 (2022). https://doi.org/10.1109/RBME.2021.3115703

4. Feldkamp, L.A., Davis, L.C., Kress, J.W.: Practical cone-beam algorithm. JOSA A **1**, 612–619 (1984)
5. HealthCare, G.: Truefidelity^{TM}: The voice of customers. Available at https://www.gehealthcare.co.uk/-/jssmedia/syneos/truefidelity/voice-of-customers/booket_truefidelity_the_voice_of_customers.pdf?rev=-1 (2023/09/16)
6. Pan, T., Hasegawa, A., Luo, D., Wu, C.C., Vikram, R.: Impact on central frequency and noise magnitude ratios by advanced CT image reconstruction techniques. Med. Phys. **47**, 480–487 (2020)
7. Racine, D., et al.: Task-based characterization of a deep learning image reconstruction and comparison with filtered back-projection and a partial model-based iterative reconstruction in abdominal CT: a phantom study. Physica Med. **76**, 28–37 (2020)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognitio, pp. 2414–2423. (2016)
9. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. Advances in Neural Inform. Process. Syst. **28** (2015)
10. Starck, J.-L., Elad, M., Donoho, D.L.: Image decomposition via the combination of sparse representations and a variational approach. IEEE Trans. Image Process. **14**(10), 1570–1582 (2005)
11. Zeng, Y., et al.: Magnetic resonance image denoising algorithm based on cartoon, texture, and residual parts. Comput. Math. Methods Med. **2020** (2020)
12. Cheng, Y., Liu, Z.: Image denoising algorithm based on structure and texture part. In: 2016 12th International Conference on Computational Intelligence and Security (CIS), pp. 147–151 (2016). https://doi.org/10.1109/CIS.2016.0042
13. Kulathilake, K., Abdullah, N., Sabri, A., Bandara, A., Lai, K.W.: A review on self-adaptation approaches and techniques in medical image denoising algorithms. Multimed. Tools Applicat. **81**, 37591 (2022)
14. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: IEEE computer society conference on computer vision and pattern recognition(CVPR'05), vol. 2, pp. 60–65 (2005)
15. Li, Z., et al.: Adaptive nonlocal means filtering based on local noise level for CT denoising. Med. Phys. **41**, 011908 (2014). https://doi.org/10.1118/1.4851635
16. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
17. Thibault, J.-B., Sauer, K., Bouman, C., Hsieh, J.: A three-dimensional statistical approach to improved image quality for multislice helical CT. Med. Phys. **34**, 4526–4544 (2007)
18. Beister, M., Kolditz, D., Kalender, W.: Iterative reconstruction methods in X-ray CT. Physica Med. **28**, 94–108 (2012)
19. Kang, E., Min, J., Ye, J.C.: A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. Med. Phys. **44**, 360–375 (2017). https://doi.org/10.1002/mp.12344
20. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016). https://doi.org/10.1109/TPAMI.2015.2439281
21. Chen, H., et al.: Low-dose CT with a residual encoder-decoder convolutional neural network. IEEE Trans. Med. Imaging **36**(12), 2524–2535 (2017)
22. Wolterink, J.M., Leiner, T., Viergever, M.A., Išgum, I.: Generative adversarial networks for noise reduction in low-dose CT. IEEE Trans. Med. Imaging **36**(12), 2536–2545 (2017)
23. Hsieh, J., Liu, E., Nett, B., Tang, J., Thibault, J.-B., Sahney, S.: A new era of image reconstruction: Truefidelity^{TM}. Tech. white paper on deep learning image reconstruction, GE Healthcare (2019)
24. Nagare, M., Melnyk, R., Rahman, O., Sauer, K.D., Bouman, C.A.: A bias-reducing loss function for CT image denoising. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (ICASSP), pp. 1175–1179 (2021). https://doi.org/10.1109/ICASSP39728.2021.9413855
25. Oh, J., Wu, D., Hong, B., Lee, D., Kang, M., Li, Q., Kim, K.: Texture-preserving low dose CT image denoising using pearson divergence. Phys. Med. Biol. **69**(11), 115021 (2024)
26. Nagare, M., et al.: A noise preserving sharpening filter for CT image enhancement. In: IEEE International Conference on Image Processing (ICIP), pp. 2566–2570 (2022)
27. Liu, J., Kang, Y., Qiang, J., Wang, Y., Hu, D., Chen, Y.: Low-dose CT imaging via cascaded resunet with spectrum loss. Methods **202**, 78–87 (2022)
28. Yang, Q., et al.: Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. IEEE Trans. Med. Imaging **37**(6), 1348–1357 (2018). https://doi.org/10.1109/TMI.2018.2827462
29. Li, M., Hsu, W., Xie, X., Cong, J., Gao, W.: SACNN: Self-attention convolutional neural network for low-dose CT denoising with self-supervised perceptual loss network. IEEE Trans. Med. Imaging **39**(7), 2289–2301 (2020)
30. You, C., et al.: CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). IEEE Trans. Med. Imaging **39**(1), 188–203 (2020). https://doi.org/10.1109/TMI.2019.2922960
31. Lee, J., Jeon, J., Hong, Y., Jeong, D., Jang, Y., Jeon, B., Baek, H.J., Cho, E., Shim, H., Chang, H.-J.: Generative adversarial network with radiomic feature reproducibility analysis for computed tomography denoising. Comput. Biol. Med. **159**, 106931 (2023). https://doi.org/10.1016/j.compbiomed.2023.106931
32. Cohen, J.P., Luck, M., Honari, S.: Distribution matching losses can hallucinate features in medical image translation. In: Medical Image Computing and Comput. Assisted Intervention (MICCAI), pp. 529–536. Springer, Cham (2018)
33. Xian, W., et al.: TextureGAN: Controlling deep image synthesis with texture patches. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8456–8465 (2018)
34. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 6840–6851 (2020)
35. Kazerouni, A., Aghdam, E.K., Heidari, M., Azad, R., Fayyaz, M., Hacihaliloglu, I., Merhof, D.: Diffusion models in medical imaging: a comprehensive survey. Med. Image Anal. **88**, 102846 (2023). https://doi.org/10.1016/j.media.2023.102846
36. Jiang, H., Imran, M., Zhang, T., Zhou, Y., Liang, M., Gong, K., Shao, W.: Fast-ddpm: Fast denoising diffusion probabilistic models for medical image-to-image generation. IEEE J. Biomed. Health Inform. (2025). https://doi.org/10.1109/JBHI.2025.3565183
37. Bromley, J., et al.: Signature verification using a "siamese" time delay neural network. Int. J. Pattern Recog. Artif. Intell. **7**, 669–688 (1993)
38. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Inform. Process. Syst., vol. 27 (2014)
39. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA (2016). http://www.deeplearningbook.org
40. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: In: International Conference on Machine learning, pp. 214–223 (2017). PMLR
41. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017)
42. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134 (2017)

43. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd Int. Conf. on Learning Representations, ICLR (2015). http://arxiv.org/abs/1412.6980
44. Abadi, M., et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org (2015). https://www.tensorflow.org/
45. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
46. Solomon, J.B., Christianson, O., Samei, E.: Quantitative comparison of noise texture across CT scanners from different manufacturers. Med. Phys. **39**, 6048–6055 (2012)
47. Yang, Q., Yan, P.: Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss. GitHub (2019)
48. Szczykutowicz, T.P., Toia, G.V., Dhanantwari, A., Nett, B.: A review of deep learning ct reconstruction: concepts, limitations, and promise in clinical practice. Curr. Radiol. Rep. **10**, 101–115 (2022). https://doi.org/10.1007/s40134-022-00399-5

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Madhuri Nagare received a B.Tech. in Electronics and Telecommunication Engineering from the College of Engineering Pune (COEP), India in 2013, an M.Tech. in Geoinformatics and Natural Resources Engineering from Indian Institute of Technology Bombay (IITB), India in 2015, and a Ph.D. from the School of Electrical and Computer Engineering, Purdue University, under the supervision of Dr. Charles A. Bouman, in 2024. She received institutional honors for graduating with the first rank in both B.Tech. and M.Tech. She gained industrial research experience while working for NEC Corporation, Tokyo, Japan (2015 - 2018) on satellite image processing, and she collaborated with GE HealthCare to develop advanced algorithms for X-ray CT image reconstruction and processing. She currently works for Apple, Inc. Her research interests are machine learning, image processing and inverse problems.

Gregery T. Buzzard is Professor of Mathematics and Director of the Center for Computational and Applied Mathematics at Purdue University, where he served as Mathematics Department Head for 7 years. He has a PhD from the University of Michigan and held postdoctoral appointments at Indiana University and Cornell University before joining Purdue. His research has led to theoretical advances in dynamical systems and experiment design and to new algorithms for image and volume reconstruction for a variety of sensing modalities. The unifying ideas in his recent work are iterative methods for image reconstruction and reduction of uncertainty through appropriate measurement schemes.

Charles A. Bouman received the B.S.E.E. degree from the University of Pennsylvania, the M.S. degree from the University of California, Berkeley, and the Ph.D. degree from Princeton University. He is currently the Showalter Professor of electrical and computer engineering and biomedical engineering with Purdue University. His research in computational imaging is focused on the integration of statistical signal processing, physics, and computation to solve problems with applications in healthcare, scientific, industrial, and consumer imaging. Dr. Bouman is a member of the National Academy of Inventors, a Fellow of the AIMBE, IS&T, and SPIE. He was the recipient of the 2023 IS&T Service Award, 2021 IEEE Signal Processing Society, Claude Shannon-Harry Nyquist Technical Achievement Award, 2014 Electronic Imaging Scientist of the Year Award, and in 2020, his paper on Plug-and-Play Priors won the SIAM Imaging Science Best Paper Prize. In 2003, he founded the IS&T Computational Imaging Conference and led the creation of IEEE Transactions on Computational Imaging.