PET Image Denoising Using a Deep Neural Network Through Fine Tuning

Kuang Gong, Jiahui Guan, Chih-Chieh Liu, and Jinyi Qi

***Abstract*—Positron emission tomography (PET) is a functional imaging modality widely used in clinical diagnosis. In this paper, we trained a deep convolutional neural network to improve PET image quality. Perceptual loss based on features derived from a pretrained VGG network, instead of the conventional mean squared error, was employed as the training loss function to preserve image details. As the number of real patient data set for training is limited, we propose to pretrain the network using simulation data and fine-tune the last few layers of the network using real data sets. Results from simulation, real brain, and lung data sets show that the proposed method is more effective in removing noise than the traditional Gaussian filtering method.**

***Index Terms*—Convolutional neural network (CNN), fine-tuning, image denoising, perceptual loss, positron emission tomography (PET).**

I. Introduction

POSITRON emission tomography (PET) is a functional imaging modality that is widely used to observe molecular-level activities inside tissues through the injection of specific radioactive tracers. Due to various physical degradation factors and limited number of detected photons, image resolution, and signal-to-noise ratio (SNR) of PET images are poor. Improving PET image quality is needed in applications, such as small lesion detection, lung cancer staging, and early diagnosis of neurological disease.

Multiple advances have been made in the past decades to improve PET SNR, such as exploiting time of flight information [1], using high-efficiency detectors with depth of interaction capability [2], extending the solid angle coverage [3], [4], and adopting more accurate system modeling in image reconstruction [5]. Various post processing methods, such as the HYPR processing [6], nonlocal mean (NLM) denoising [7], [8], and anatomical guided methods [9], [10], have also been developed.

Recently, deep neural networks (DNNs) have found successful applications in various computer vision tasks, such as image segmentation [11], object detection [12], and image super resolution [13], by demonstrating better performance than the state-of-the-art methods when a large amount of training data are available. DNNs, using either convolutional neural network (CNN) [14]–[16] or generative adversarial network [17], have also been applied to medical image denoising, and showed comparable or superior results to the traditional iterative reconstruction but at a faster speed. Most of the denoising studies use images generated from high dose or fully sampled data sets as training labels, and images from low dose or partially sampled data as training inputs. Mean squared error (MSE) between the network outputs and training labels is often employed as the training loss function. There exist two issues in the application of DNN to PET image denoising. One is the lack of sufficient number of label images for training. The other is that MSE-based loss function often results in blurry network outputs [18]–[20].

In this paper, we apply DNN to PET image denoising and propose solutions to address these issues. First, to generate label images for training, we sum an hour-long dynamic PET scan into a high-count frame and use the reconstructed image as a label. The corresponding noisy input is obtained by down-sampling the high-count data to a lower count level and reconstructing the resulting low-count data. Since the number of real patient data sets is limited, we propose to pretrain the neural network using computer simulated data and then fine-tune the network using real data sets. A similar idea was presented in [21], where an MRI denoising network was first trained using CT images and then fine-tuned by MRI images. To address the blurry problem of MSE loss function, perceptual loss, which was calculated based on features extracted from a pretrained network [18], was adopted as the training loss function. Since the perceptual loss is feature-based, it can preserve more image details than the MSE loss function. The idea is similar to the one used in the anatomically constrained network [22], where features were extracted from the hidden layer of an auto-encoder.

II. Method

A. Convolutional Neural Network

The basic unit of a CNN contains a convolution layer and an activation layer. The input and output relationship of the ith unit can be described by

$$\boldsymbol{y}_i = f_i(\boldsymbol{y}_{i-1}) = g(\boldsymbol{w}_i \circledast \boldsymbol{y}_{i-1} + \boldsymbol{b}_i) \tag{1}$$

Manuscript received March 16, 2018; revised July 19, 2018 and September 19, 2018; accepted October 16, 2018. Date of publication October 23, 2018; date of current version March 1, 2019. This work was supported by the National Institutes of Health under Grant R01EB000194. *(Corresponding author: Jinyi Qi.)*

K. Gong, C.-C. Liu, and J. Qi are with the Department of Biomedical Engineering, University of California at Davis, Davis, CA 95616 USA (e-mail: qi@ucdavis.edu).

J. Guan is with the Department of Statistics, University of California at Davis, Davis, CA 95616 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRPMS.2018.2877644

where $\boldsymbol{y}_{i-1} \in \mathbb{R}^{N\times N\times C}$ denotes the unit input with spatial size $N \times N$ and C channels, $\boldsymbol{y}_i \in \mathbb{R}^{N\times N\times H}$ is the unit output with spatial size $N \times N$ and H channels, $\boldsymbol{w}_i \in \mathbb{R}^{M\times M\times C\times H}$ is the convolutional filter with kernel width M, $\boldsymbol{b} \in \mathbb{R}^{1\times H}$ is the bias term, $\circledast$ indicates the convolution operation, and g represents the nonlinear activation function. In this paper, we use the rectified linear unit (ReLU) activation function, defined as

$$g(\boldsymbol{x}) = \max(\boldsymbol{x}, \boldsymbol{0}). \tag{2}$$

To stabilize and accelerate the deep network training, batch normalization [23] is added after the convolution operation. After stacking L units together, the network output can be written as

$$\boldsymbol{y}_{\text{out}} = f_L\left(f_{L-1}\left(\ldots f_1\left(\boldsymbol{x}_{\text{input}}\right)\right)\right). \tag{3}$$

For PET image denoising, $\boldsymbol{x}_{\text{input}}$ is an noisy image reconstructed from a low-count data set, and $\boldsymbol{y}_{\text{out}}$ is the denoised PET image with improved SNR. The ability of a neural network to approximate the mapping from a noisy image to the corresponding training label is dependent on the network depth (number of layers) and structure. Deeper networks can have higher capability, but at a cost of requiring more training samples and longer training time.

B. Perceptual Loss

In most previous works, MSE between the training label $\boldsymbol{y}_{\text{label}}$ and the network output $\boldsymbol{y}_{\text{out}}$ was used as the loss function. It is defined as

$$L_{\text{mse}} = \left\|\boldsymbol{y}_{\text{label}} - \boldsymbol{y}_{\text{out}}\right\|_2^2. \tag{4}$$

It has been observed that MSE-based loss often produced blurry network outputs [18]–[20]. To preserve image details, we propose to use the perceptual loss as the objective function, which is calculated by

$$L_{\text{perceptual}} = \left\|\phi\left(\boldsymbol{y}_{\text{label}}\right) - \phi\left(\boldsymbol{y}_{\text{out}}\right)\right\|_2^2 \tag{5}$$

where ϕ represents the feature extraction operator and is based on the intermediate layer output from a pretrained network. By comparing feature maps instead of pixel intensities, the network can be more effective in removing noise while keeping image details. In this paper, we adopted the output before the first pooling layer from the VGG19 network [24] as the extracted features. The VGG19 network architecture contains 16 convolutional layers followed by three fully connected layers. The VGG network was trained using ImageNet, which is a large database of natural images [25]. A total of 64 feature maps were extracted with the same spatial size as the input. This process is illustrated in Fig. 1. We hypothesize that the low-level features trained from natural images are also present in medical images. We have tried to use the features extracted from deeper layers, but the performance is not as good as that of the first layer. The reason for this is worth further investigation.

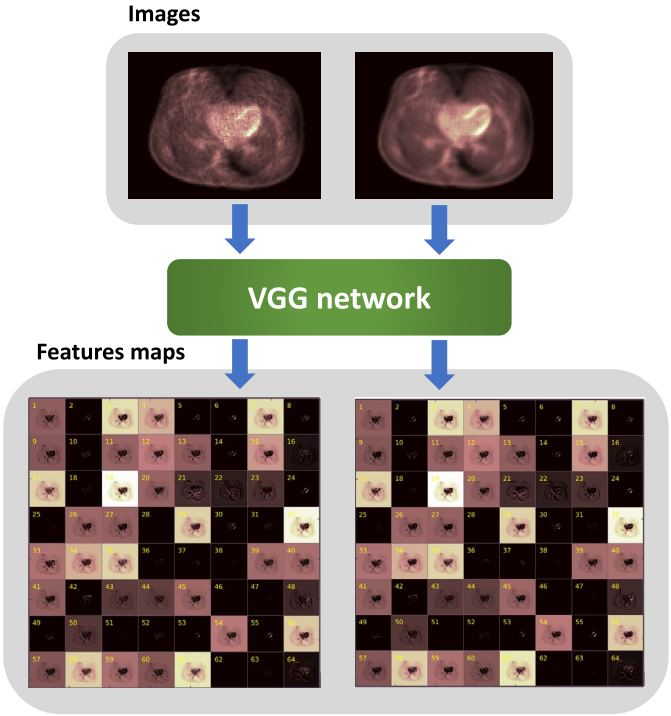

Fig. 1. Schematic of the feature map generation process based on the VGG network. Top rows are the input images and bottom rows are the feature maps extracted from the VGG network. Left column is the image reconstructed from low-count data and right column is the image reconstructed from high-count data.

C. Network Structure

Our network structure is similar to the residual neural network used in [19]. A schematic of the network architecture is shown in Fig. 2. The network consists of a cascade of five residual blocks [26]. Each residual block contains two repetitions of a 3×3 convolutional layer, a batch normalization layer, and a ReLU layer. Skip connection is added between the start and the end of each block. Another skip connection is added between the first and last stages of the whole network. The number of features for each convolutional layer is 64, and the spatial size of the network input is 128×128. Five input channels are used to include the center slice as well as four neighboring axial slices for effective noise removal and reduction of axial artifacts.

D. Fine-Tuning

Due to limited number of real data sets for training, we propose to pretrain the network using simulated data first and then fine-tune the network using real data. Compared with real data sets, simulated data sets are much easier to generate. Using realistic phantoms and an accurate physical model of the PET scanner, simulated data sets can have high similarity to real data sets, which can facilitate the fine tuning. This framework also allows continuous improvement of the network by incorporating new patient data. Since the front layers generally extracts low-level image features that are common to different types of images, we only fine-tune the last few layers in the red shadow region in Fig. 2. In addition, the batch normalization layers of the whole network were also

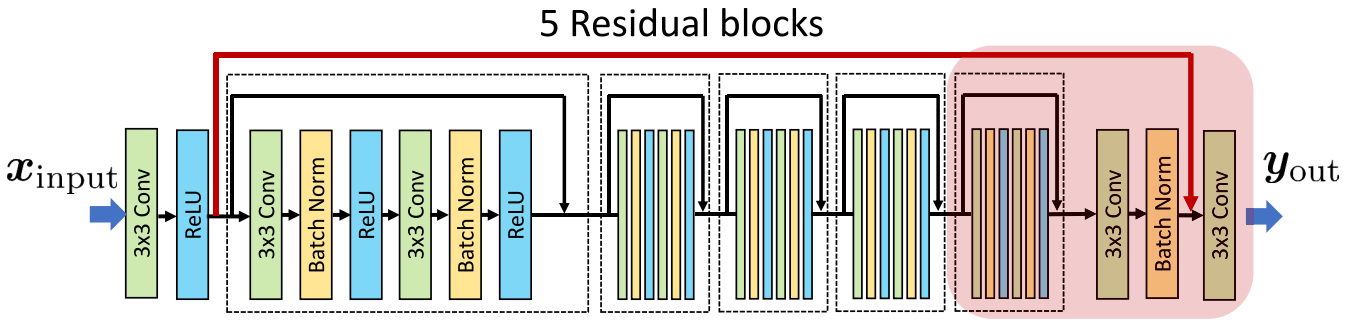

Fig. 2. Schematic of the neural network architecture. The red shadow region indicates the layers that are fine tuned by real data.

fine-tuned as the intensity levels can be different between the simulation and real data sets. We separately trained two networks, one for brain imaging and one for lung imaging. The brain-imaging network was pretrained using brain phantoms from the BrainWeb [27] and the lung-imaging network was pretrained using the XCAT phantom [28].

III. Experimental Setup

A. Brain Phantom Simulation

Nineteen 3-D brain phantoms from BrainWeb [27] were employed in the simulation. Eighteen phantoms were used for training and one phantom was reserved for testing. The computer simulation modeled the geometry of a Siemens mCT scanner [29]. The system matrix was modeled by using the multiray tracing method [30]. The image array size was $128\times128\times105$ and the voxel size was $2\times2\times2$ mm^3. The time activity curves of blood, gray matter, and white matter were the same as those used in [31] to mimic an FDG scan. Noise-free sinogram data were generated by forward-projecting the ground-truth images using the system matrix and the attenuation map. Poisson noise was then introduced to the noise-free data after scaling the total counts to the level of a 1-h FDG scan with 5 mCi injection. Uniform random events were simulated and accounted for 30% of the noise-free data. Scatters were not included.

For network training, each 1-h scan was summed into one frame and reconstructed as the label, and the noisy input was obtained by down-sampling the 1-h data to 1/5th of counts and reconstructing the low-count data. All images were reconstructed using ML EM with 120 iterations. A total of 18 (number of phantoms) $\times$ 75 (number of axial slices extracted from each phantom) training image pairs were generated after discarding axial slices at the two ends with little activity. Examples of training images are shown in Fig. 3.

For testing, the last 10-min static frame was extracted from the 1-h scan and reconstructed as the noisy input. The 10-min static frame has similar count level as the training input. The CNN denoised images were compared with those obtained by traditional Gaussian smoothing and NLM denoising. For quantitative evaluation, contrast recovery coefficient (CRC) versus the standard deviation (STD) curves were calculated based on reconstructions of 20 independent and identically distributed (i.i.d) realizations. The CRC was computed between selected

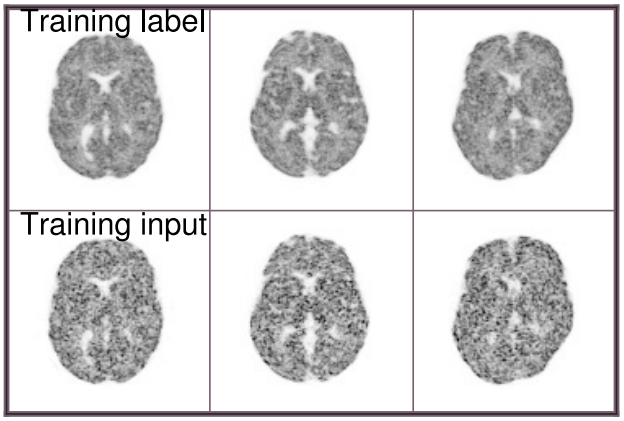

Fig. 3. Three pairs of the training images from the simulated brain phantom data. Top row contains the training labels and bottom row contains the corresponding noisy inputs.

gray matter regions and background white matter regions as

$$\text{CRC} = \frac{1}{R}\sum_{r=1}^{R}\left(\frac{\bar{a}_r}{\bar{b}_r} - 1\right)\Big/\left(\frac{a^{\text{true}}}{b^{\text{true}}} - 1\right) \tag{6}$$

where $\bar{a}_r = 1/K_a \sum_{k=1}^{K_a} a_{r,k}$ is the average uptake over $K_a = 12$ gray matter ROIs in realization r, $\bar{b}_r = 1/K_b \sum_{k=1}^{K_b} b_{r,k}$ is the average value of the background ROIs in realization r, and R is the number of realizations. The background STD was computed as

$$\text{STD} = \frac{1}{K_b}\sum_{k=1}^{K_b}\frac{\sqrt{\frac{1}{R-1}\sum_{r=1}^{R}(b_{r,k} - \bar{b}_k)^2}}{\bar{b}_k} \tag{7}$$

where $\bar{b}_k = 1/R \sum_{r=1}^{R} b_{r,k}$ is the average of the kth background ROI means over realizations and K_b is the number of background ROIs. When choosing the gray matter ROIs, only those pixels inside predefined 20-mm-diameter spheres and containing 80% of gray matter were included. Background ROIs consist of 37 circular regions with a diameter of 12 mm drawn in the white matter region.

B. Real Brain Data Sets

After pretraining the network using BrainWeb phantoms, we fine-tuned the network using real data from a brain PET scanner [32]. Two dynamic brain PET scans of 70 min with 5 mCi FDG injection were used for the fine-tuning and

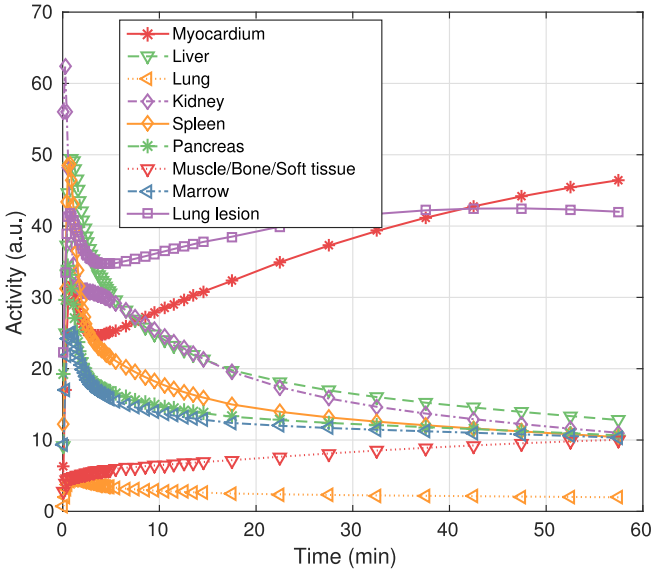

Fig. 4. TAC curves of different organs and lesions used in the XCAT simulation.

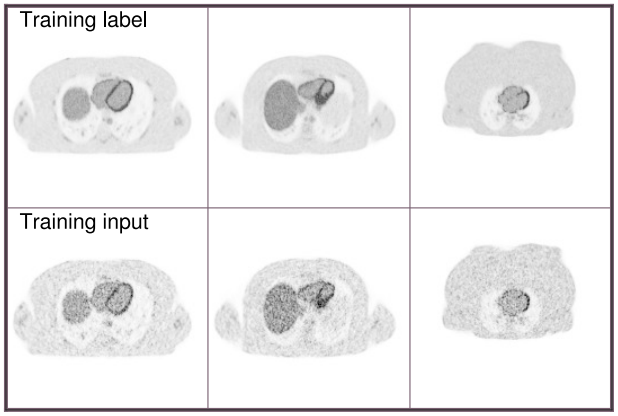

Fig. 5. Three pairs of the training images from the XCAT phantom simulation. Top row contains the training labels and bottom row contains the corresponding noisy inputs.

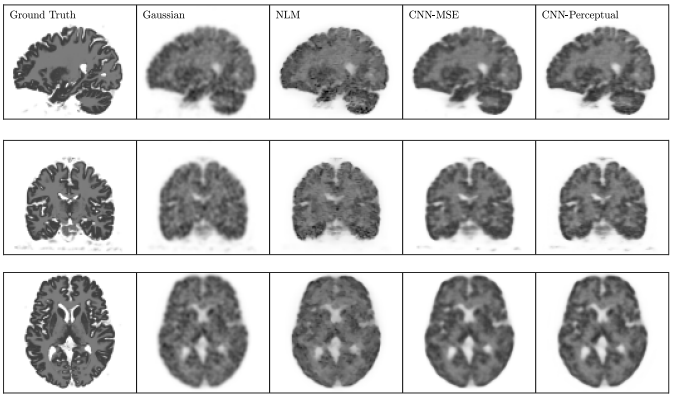

Fig. 6. Three orthogonal slices of the reconstructed last-10-min static frame of the test phantom. First column: ground truth; second column: EM images smoothed by Gaussian filtering; third column: EM images smoothed by NLM denoising; fourth column: EM images with CNN using MSE loss; and fifth column: EM images with CNN using perceptual loss. The images were selected by matching the background noise level (see Fig. 7).

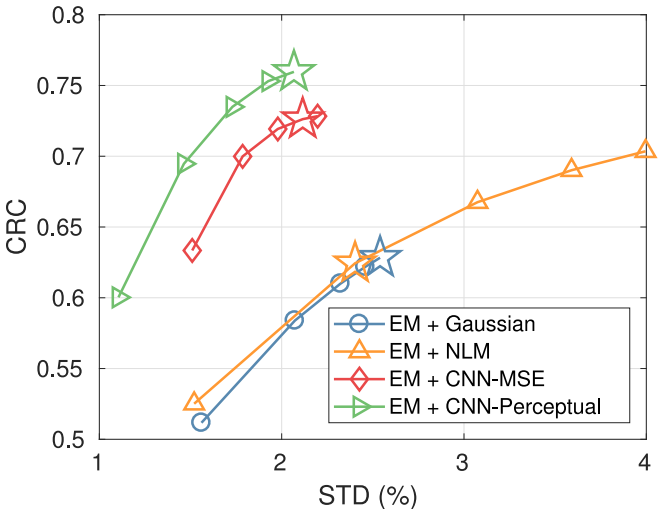

Fig. 7. CRC-STD curves of the denoised images for the last-10-min static frame of the test BrainWeb phantom. Markers are plotted every 24 iterations with the lowest point corresponding to the 24th iteration. The images shown in Fig. 6 are labeled by ⋆ markers.

another patient dataset was reserved for testing. In fine-tuning, images reconstructed using the whole 70 min scan were treated as the training labels and images from 1/5th of the counts were used as the training inputs. All image reconstructions were performed using the ML EM algorithm with 120 iterations. Correction factors for randoms, scatters were included in the forward model during reconstruction. Attenuation was derived from a T1-weighted MR image using the SPM-based atlas method [33]. The reconstructed image array size was $256\times256\times153$ and the voxel size was $1.25\times1.25\times1.25$ mm^3. Two 128×128 patches were randomly extracted from each reconstructed image slice for fine-tuning. As the patch extraction is a random process, there might be overlapping between extracted patches. A total of 520 training pairs (two training data sets, each containing 130 axial slices and each axial slice generating two patches) were extracted. For comparison, we also trained the network using the real data directly without the pretraining stage. During testing, network input spatial size was set to 256×256 so that each image can be processed directly without splitting. As the ground truth of the real data is unknown, a hot sphere of diameter 12.5 mm, mimicking a tumor, was added to the test sinogram data. The TAC of the hot sphere as added to the background was set to the TAC of the gray matter, so the final TAC of the simulated tumor region is higher than that of the gray matter because of the superposition. Twenty i.i.d realizations of low-count test data were generated and reconstructed. Images with and without the inserted tumor were reconstructed and the difference was taken to obtain the tumor only image. The tumor contrast recovery (CR) was calculated as

$$\mathrm{CR} = \frac{1}{R}\sum_{r=1}^{R}\bar{l}_r/l_{\mathrm{true}} \tag{8}$$

where $\bar{l}_r$ is the mean tumor intensity inside the tumor ROI, l_{true} is the ground truth of the tumor intensity, and R is the number of the realizations. For the background, 23 circular ROIs with

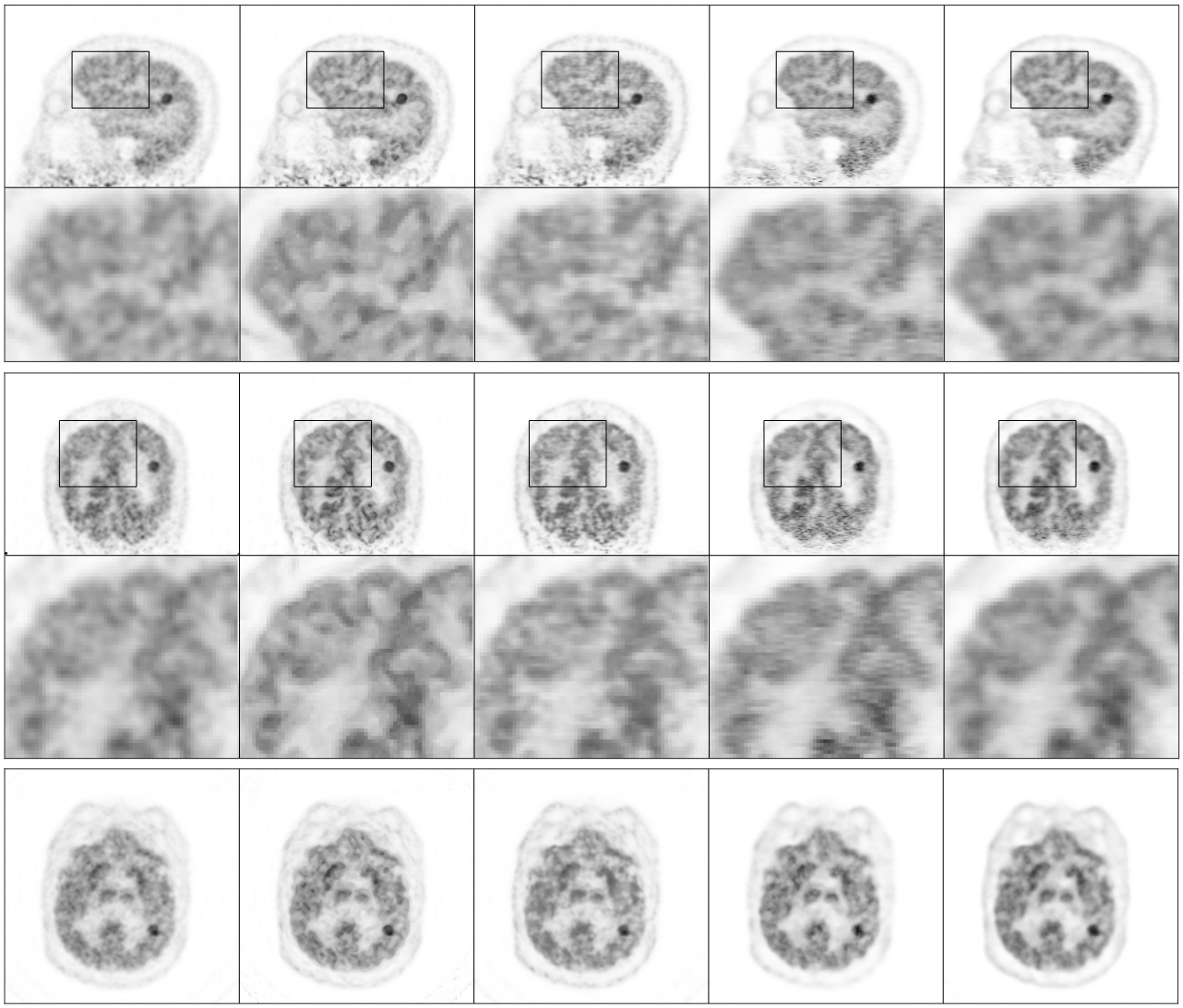

Fig. 8. Three orthogonal views of the reconstructed real brain test data set using different methods. Two cortex regions from the sagittal and coronal views are zoomed in for easier visual comparison. First column: EM image smoothed by Gaussian denoising; second column: EM images smoothed by NLM denoising; third column: EM image denoised by CNN trained from simulated phantom; fourth column: EM image denoised by CNN from real data only; and fifth column: EM image denoised by CNN with fine-tuning. The images were selected by matching the background noise level (see Fig. 9).

a diameter of 5 mm were drawn in the white matter and the STD was calculated according to (7).

C. Lung Phantom Simulation

To pretrain a network for lung imaging, 1-h scan of 19 XCAT phantoms [28] with different organ sizes and genders were simulated. Eighteen phantoms were used for training and one phantom was reserved for testing. Apart from the major organs, 30 hot spheres of diameters ranging from 12.8 to 22.4 mm were inserted into the training phantoms as lung lesions. For the test image, five lesions with diameter 16.35 mm were inserted. Two-tissue-compartment model mimicking an FDG scan with analytical blood input function was used to generate the time activities [34]. In order to simulate population differences, each kinetic parameter was modeled as a Gaussian variable with coefficient of variation equal to 0.1. Mean of the time activities for different organs and lung lesions are shown in Fig. 4. The scanner geometry mimics a GE 690 scanner [35]. Uniform random and scatter events were simulated and accounted for 60% of the noise free prompt data to match those observed in real data sets. Poisson noise was added mimicking a 5-mCi FDG injection. Images reconstructed using counts from the last 40 min were treated as the training labels and images using one-tenth of the 40-min counts as the training inputs. All image reconstructions were performed using the ML EM algorithm with 100 iterations. Three training pairs from different phantoms are shown in Fig. 5. The image matrix size is $128 \times 128 \times 49$ and the voxel size is $3.27 \times 3.27 \times 3.27$ mm^3. A total of 18 (number of phantoms) $\times$ 49 (number of axial slices extracted from each phantom) training image pairs were generated. For testing, the last 5-min static frame was extracted from the 1-h scan and reconstructed as the noisy input. The 5-min static frame has similar count level as the training input. The lesion CR was calculated according to (8). Forty-two background ROIs

were chosen in the liver region to calculate the STD according to (7).

D. Real Lung Data Sets

For fine-tuning, five patient data sets (1-h FDG dynamic scan with 5 mCi injection) acquired on a GE 690 scanner were employed in the training and another patient data set was reserved for testing. Normalization, attenuation correction, randoms, and scatters were generated using the manufacturer software and included in image reconstruction. Images reconstructed using counts from the last 40 min were treated as the training labels and images using one-tenth of the 40-min counts as the training inputs. All image reconstructions were performed using the ML EM algorithm with 100 iterations. A total of 5 (number of patient data sets) $\times$ 49 (number of axial slices extracted from each patient data set) training image pairs were generated. Spherical lesions with a diameter of 12.8 mm were inserted in the testing sinograms for quantitative analysis. The TAC of the lesions inserted was similar to the TAC of the liver, so the final TAC of the simulated lesion region is higher than that of the liver because of the superposition. A total of 20 i.i.d realizations of test data were generated by randomly sampling one-tenth of the last 40-min counts and reconstructed. For lesion quantification, images with and without the inserted lesion were reconstructed and the difference was taken to obtain the lesion only image. The lesion CR was calculated according to (8). Forty-seven background ROIs were chosen in the liver region to calculate the STD according to (7).

E. Implementation Details

The proposed neural network was implemented using TensorFlow 1.4, which is a deep learning platform with back-propagation implemented using automatic differentiation. The Adam algorithm, which is a popular adaptive stochastic gradient method [36], was used as the optimizer. The learning rate and the decay rates used the default settings in TensorFlow. Perceptual loss was used in all CNN training unless noted otherwise. All training and fine-tuning used a batch size of 30 and 500 epochs. Gaussian filtering and NLM denoising were used as the reference methods. The full-width-half-maximum of the Gaussian filter was 1.5 voxels in all cases. For the NLM method, the patch size was $3\times3\times3$, and the searching window size was $5\times5\times5$. The STD of the NLM Gaussian weighting function was set to be the STD of the image. These parameters were chosen empirically to optimize the contrast versus noise tradeoff. The CRC/CR-STD curves were generated by varying the ML EM iteration number.

IV. Results

Fig. 6 shows the denoised results of the last 10-min static frame of the simulated brain phantom data. We can see that compared with the result using the Gaussian filter, the CNN denoised images preserve more details of the brain structure and also has higher contrast between the gray matter and white matter. The CRC of the gray matter versus STD of the white matter curves are plotted in Fig. 7 by varying the EM iteration number. We can see that the CNN denoising provides much better CRC versus STD tradeoff than the Gaussian and NLM filters. Comparing with the CNN (same network structure) trained using MSE loss, the CNN trained using the perceptual loss achieves a higher CRC at any matched STD level.

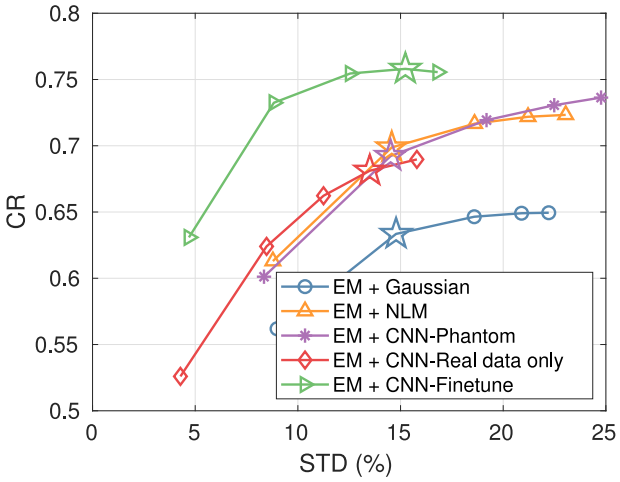

Fig. 9. CR-STD curves for the real brain test data set denoised using different methods. Markers are plotted every 24 iterations with the lowest point corresponding to the 24th iteration. The images shown in Fig. 8 are labeled by $\star$ markers.

For the real brain data sets, denoised images of one low-count realization are shown in Fig. 8. We can see that after applying the CNN method, cortical boundary becomes clearer and the image noise is reduced. Also the result using CNN with fine-tuning is sharper and less noisy than the results without fine-tuning, which indicates the effectiveness of pre-training plus fine-tuning. Fig. 9 shows the CR-STD curves, which confirm that the CNN with fine-tuning has the best CR-STD tradeoffs.

Fig. 10 shows the denoised results of the last 5-min static frame of the simulated XCAT phantom data. We can see that the neural network denoising methods result in lower noise than the Gaussian and NLM denoising methods. Compared with the CNN trained using MSE loss, the CNN trained with perceptual loss generates images with higher contrast in the lesion and myocardium region. The curves of the CR of the inserted lung lesion versus STD in the liver region are plotted in Fig. 11, which further confirms our observation.

Fig. 12 shows the reconstructed images of a lung testing data set. Here, we also included the denoising results using the CNN trained by phantom data only. We can see that the CNN methods result in clearer details in the spinal regions and also lower noise compared with the Gaussian and NLM denoising methods. Also the contrast of the inserted lesion is higher in the CNN result with fine-tuning than those from CNN trained with either real data or phantom data only. Fig. 13 compares the CR-STD curves. It shows that the CNN method with fine-tuning has the best performance—it provides a nearly twofold STD reduction as compared with the Gaussian denoising method. By comparing between different CNN denoising results, we can clearly see the benefits of fine-tuning.

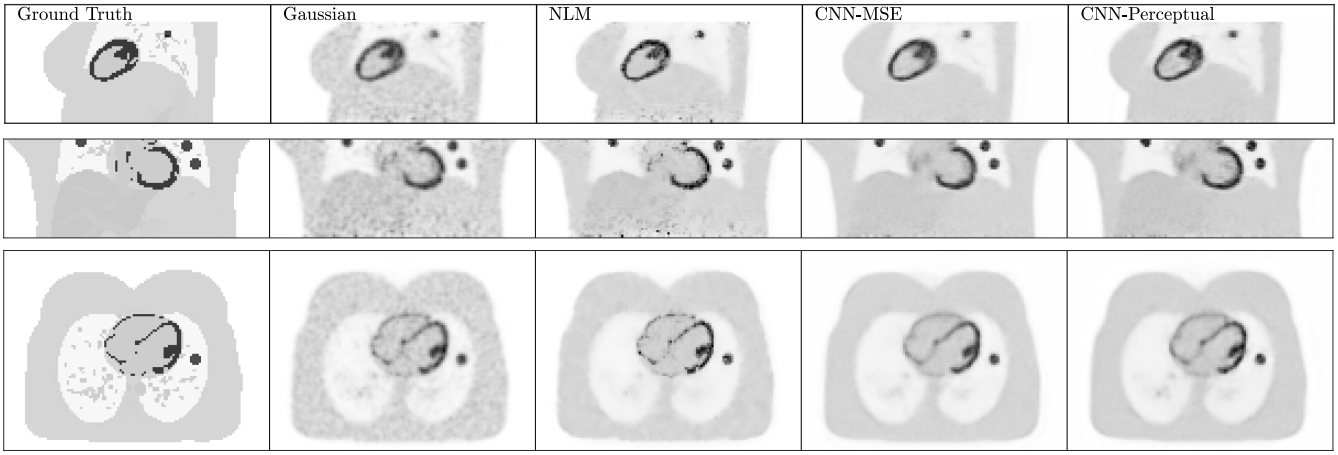

Fig. 10. Three orthogonal slices of the reconstructed last-5-min static frame of the test XCAT phantom. The locations of these images in CR-STD plots are marked by ⋆ in Fig. 11. First column: ground truth; second column: EM images smoothed by Gaussian filtering; third column: EM images smoothed by NLM denoising; fourth column: EM images with CNN using MSE loss; and fifth column: EM images with CNN using perceptual loss.

V. Discussion

Deep neural network can learn a complex relationship between the input and output provided that a large amount of training data is available. However, training a deep network from scratch with a limited amount of data can lead to inferior performance due to overfitting, as demonstrated by the real data studies in this paper. Pretraining followed by fine-tuning is an effective technique to address this issue, because features extracted at the early stages can be shared. The benefit of fine-tuning is clearly demonstrated by comparing the CNN denoising results with and without fine tuning. Our results also show that CNN trained by phantom data can be applied to real data when the simulation models the real imaging condition, but the performance is worse than the CNN fine-tuned with real data. In the simulation, we used a precomputed forward projector to generate data. The forward projector modeled the solid angle effect and crystal penetration, but not intercrystal scattering. If simulation data were generated by a more accurate Monte Carlo simulation, such as GATE [37], the results of the phantom-only CNN might be improved.

In this paper, we used the images reconstructed from 60-min or 40-min long data sets as the training label and the images reconstructed from down-sampled data sets as the training input. While the long scans may have different contrast from standard static scans, our simulation results have shown that the learned neural network can be applied to short static scans with a matched noise level. More quantitative evaluations using clinical data sets are needed for further evaluation. During the experiments, we found that the best performance of the neural network was achieved when the noise level of the testing data was similar to the training data. If there was a mismatch between the training and testing data noise levels, the network performance would be degraded. One explanation is that if the noise level is different, then there is a large chance that the testing data do not lie in the training data space. Hence, to have the best improvement using neural network methods, a new training session is recommended if the noise level of the test data is outside of the training noise level.

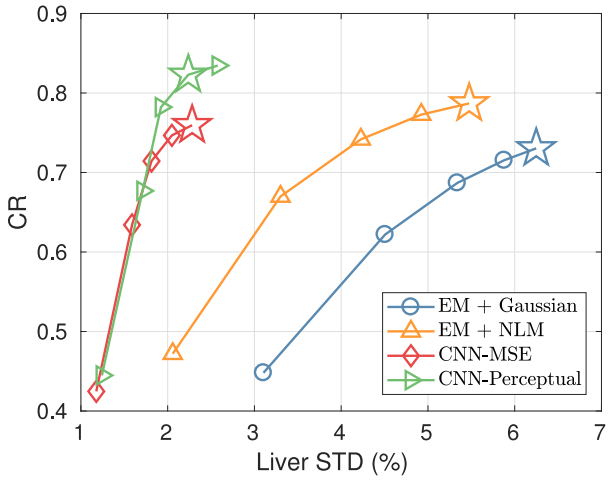

Fig. 11. CR-STD curves of the denoised images for the last-5-min static frame of the test XCAT phantom. Markers are plotted every 20 iterations with the lowest point corresponding to the 20th iteration. The images shown in Fig. 10 are labeled by ⋆ markers.

When we designed the experiments, we wanted to test the effect of pretraining using phantoms in two cases: 1) simulation settings are different from the real datasets and 2) simulation settings are almost the same as the real datasets. In the lung simulation, we made the simulation to be similar to the real data as much as we can. For the brain study, the simulation and real data settings (in terms of image pixel size, scanner geometry, etc.) were chosen to be different to test whether fine-tuning is still useful when the phantom study and the later patient study do not match. In both cases, we found that the image output of CNN with fine-tuning is better than those of CNNs trained using either simulation or real data alone. This result is encouraging as it indicates that we may be able to combine real data from different scanners to increase the number of training images in practice.

One limitation of our network is that 2-D convolution was used. To exploit information along the axial dimension, five input channels were utilized to include neighboring axial slices. Alternatively, 3-D convolution can be used and may

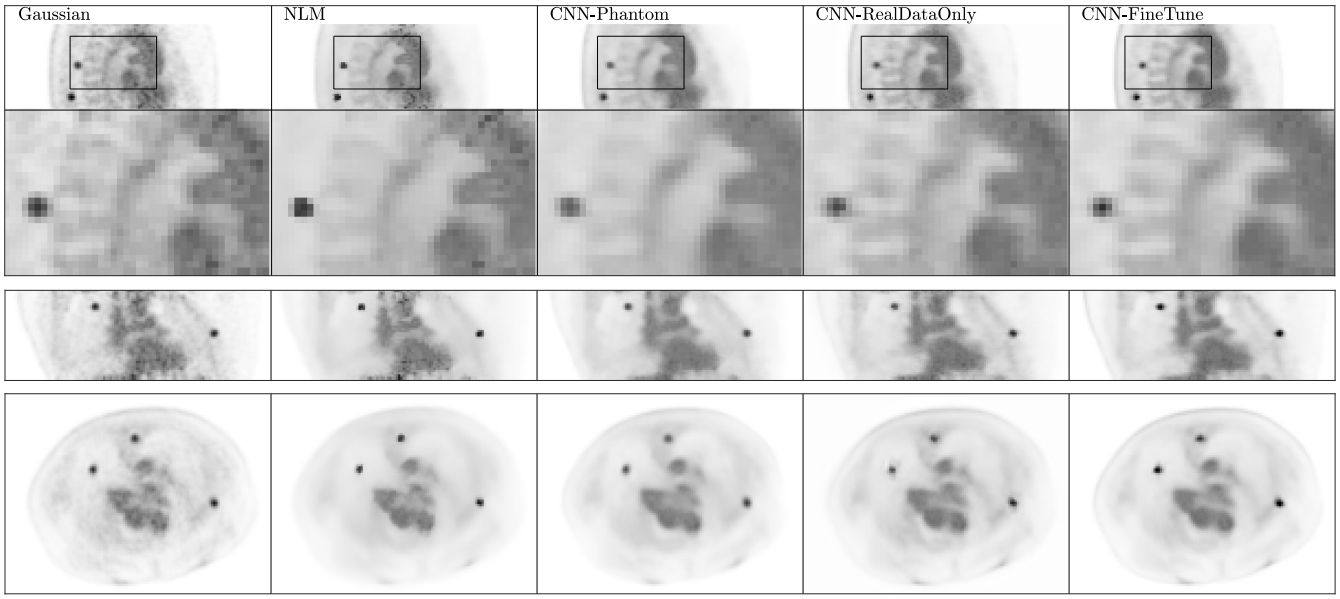

Fig. 12. Three orthogonal views of the reconstructed real lung test data set using different methods. Spine regions in the sagittal view are zoomed in for easier visual comparison. The locations of these images in CR-STD plots are shown as ⋆ markers in Fig. 13. First column: EM image smoothed by Gaussian denoising; second column: EM images smoothed by NLM denoising; third column: EM image denoised by CNN trained from simulated phantom; fourth column: EM image denoised by CNN from real data only; and fifth column: EM image denoised by CNN with fine-tuning.

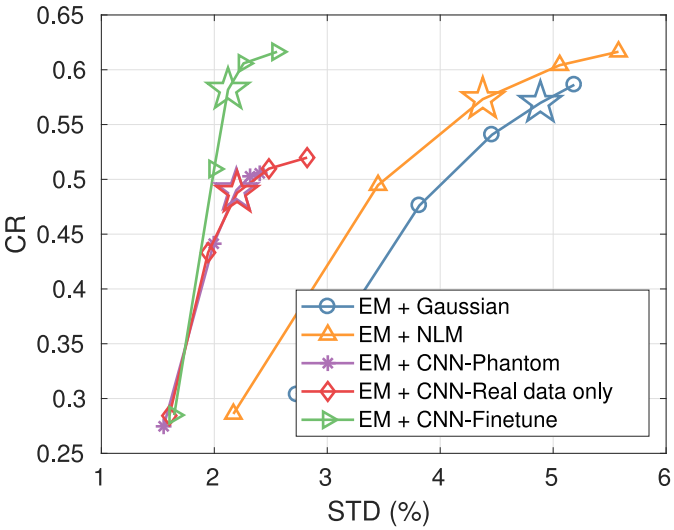

Fig. 13. CR-STD curves for the real lung test data set denoised using different methods. Markers are plotted every 20 iterations with the lowest point corresponding to the 20th iteration. The images shown in Fig. 12 are labeled by ⋆ markers.

be able to extract more axial information than using multiple input channels because axial information is preserved at all layers. Extension to 3-D convolutional network will be investigated in our future work.

VI. Conclusion

In this paper, we have applied a deep neural network to PET image denoising based on perceptual loss. The proposed pre-training plus fine-tuning strategy can help to train a deep neural network with limited amount of real data. Both simulation and real data experiments show that the proposed framework can produce images with better quality than post-smoothing using a Gaussian or NLM filter. Further work will focus on exploring 3-D networks as well as more real data evaluations.

Acknowledgment

The authors would like to thank Dr. M. Judenhofer at UC Davis for sharing GPU resources and Dr. C. Catana at MGH for sharing the dynamic brain data sets.

References

[1] J. S. Karp, S. Surti, M. E. Daube-Witherspoon, and G. Muehllehner, "Benefit of time-of-flight in PET: Experimental and clinical results," *J. Nucl. Med.*, vol. 49, no. 3, pp. 462–470, 2008.

[2] Y. Yang *et al.*, "Depth of interaction calibration for PET detectors with dual-ended readout by PSAPDs," *Phys. Med. Biol.*, vol. 54, no. 2, pp. 433–445, 2009.

[3] J. K. Poon *et al.*, "Optimal whole-body PET scanner configurations for different volumes of LSO scintillator: A simulation study," *Phys. Med. Biol.*, vol. 57, no. 13, pp. 4077–4094, 2012.

[4] K. Gong *et al.*, "Designing a compact high performance brain PET scanner-simulation study," *Phys. Med. Biol.*, vol. 61, no. 10, pp. 3681–3697, 2016.

[5] K. Gong *et al.*, "Sinogram blurring matrix estimation from point sources measurements with rank-one approximation for fully 3-D PET," *IEEE Trans. Med. Imag.*, vol. 36, no. 10, pp. 2179–2188, Oct. 2017.

[6] B. T. Christian, N. T. Vandehey, J. M. Floberg, and C. A. Mistretta, "Dynamic PET denoising with HYPR processing," *J. Nucl. Med.*, vol. 51, no. 7, pp. 1147–1154, 2010.

[7] J. Dutta, R. M. Leahy, and Q. Li, "Non-local means denoising of dynamic PET images," *PLoS ONE*, vol. 8, no. 12, 2013, Art. no. e81390.

[8] C. Chan, R. Fulton, R. Barnett, D. D. Feng, and S. Meikle, "Postreconstruction nonlocal means filtering of whole-body PET with an anatomical prior," *IEEE Trans. Med. Imag.*, vol. 33, no. 3, pp. 636–650, Mar. 2014.

[9] H. Wang and B. Fei, "An MR image-guided, voxel-based partial volume correction method for PET images," *Med. Phys.*, vol. 39, no. 1, pp. 179–195, 2012.

[10] J. Yan, J. C.-S. Lim, and D. W. Townsend, "MRI-guided brain PET image filtering and partial volume correction," *Phys. Med. Biol.*, vol. 60, no. 3, pp. 961–976, 2015.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, Munich, Germany, 2015, pp. 234–241.
[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 91–99.
[13] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[14] S. Wang *et al.*, "Accelerating magnetic resonance imaging via deep learning," in *Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI)*, Prague, Czech Republic, 2016, pp. 514–517.
[15] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction," *Med. Phys.*, vol. 44, no. 10, pp. e360–e375, 2017, doi: 10.1002/mp.12344.
[16] H. Chen *et al.*, "Low-dose CT via convolutional neural network," *Biomed. Opt. Exp.*, vol. 8, no. 2, pp. 679–694, 2017.
[17] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Generative adversarial networks for noise reduction in low-dose CT," *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2536–2545, Dec. 2017.
[18] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
[19] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. CVPR*, vol. 2, 2017, p. 4, doi: 10.1109/CVPR.2017.19.
[20] Q. Yang, P. Yan, M. K. Kalra, and G. Wang, "CT image denoising with perceptive deep neural networks," *arXiv preprint arXiv:1702.07019*, 2017. [Online]. Available: https://arxiv.org/abs/1702.07019
[21] Y. Han *et al.*, "Deep learning with domain adaptation for accelerated projection-reconstruction MR," *Magn. Reson. Med.*, vol. 80, no. 3, pp. 1189–1205, 2018, doi: 10.1002/mrm.27106.
[22] O. Oktay *et al.*, "Anatomically constrained neural networks (ACNN): Application to cardiac image enhancement and segmentation," *IEEE Trans. Med. Imag.*, vol. 37, no. 2, pp. 384–395, Feb. 2018.
[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 448–456.
[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556
[25] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 248–255.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.
[27] B. Aubert-Broche, M. Griffin, G. B. Pike, A. C. Evans, and D. L. Collins, "Twenty new digital brain phantoms for creation of validation image data bases," *IEEE Trans. Med. Imag.*, vol. 25, no. 11, pp. 1410–1416, Nov. 2006.
[28] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. Tsui, "4D XCAT phantom for multimodality imaging research," *Med. Phys.*, vol. 37, no. 9, pp. 4902–4915, 2010.
[29] B. W. Jakoby *et al.*, "Physical and clinical performance of the mCT time-of-flight PET/CT scanner," *Phys. Med. Biol.*, vol. 56, no. 8, pp. 2375–2389, 2011.
[30] J. Zhou and J. Qi, "Fast and efficient fully 3D PET image reconstruction using sparse system matrix factorization with GPU acceleration," *Phys. Med. Biol.*, vol. 56, no. 20, pp. 6739–6757, 2011.
[31] K. Gong *et al.*, "Direct Patlak reconstruction from dynamic PET data using the kernel method with MRI information based on structural similarity," *IEEE Trans. Med. Imag.*, vol. 37, no. 4, pp. 955–965, Apr. 2018.
[32] A. Kolb *et al.*, "Technical performance evaluation of a human brain PET/MRI system," *Eur. Radiol.*, vol. 22, no. 8, pp. 1776–1788, 2012.
[33] D. Izquierdo-Garcia *et al.*, "An spm8-based approach for attenuation correction combining segmentation and nonrigid template formation: Application to simultaneous PET/MR brain imaging," *J. Nucl. Med.*, vol. 55, no. 11, pp. 1825–1830, 2014.
[34] K. Gong *et al.*, "Iterative PET image reconstruction using convolutional neural network representation," *IEEE Trans. Med. Imag.*, to be published, doi: 10.1109/TMI.2018.2869871.
[35] V. Bettinardi *et al.*, "Physical performance of the new hybrid PET/CT discovery-690," *Med. Phys.*, vol. 38, no. 10, pp. 5394–5411, 2011.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980
[37] S. Jan *et al.*, "Gate V6: A major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy," *Phys. Med. Biol.*, vol. 56, no. 4, pp. 881–901, 2011.