Targeted Gradient Descent: A Novel Method for Convolutional Neural Networks Fine-tuning and Online-learning

Junyu Chen[1,2,*], Evren Asma[3], Chung Chan[3]

[1]Department of Radiology and Radiological Science, Johns Hopkins University, MD, USA

[2]Department of Electrical and Computer Engineering, Johns Hopkins University, MD, USA

[3]Canon Medical Research USA, INC., IL, USA

*: This work was done while the author was an intern at Canon Medical Research USA

Introduction

When deploying a pretrained ConvNet for clinical applications, we often face two challenges:

- When new imaging systems and or updated reconstruction algorithms are employed:
  - Image quality and appearance will change.
  - Neural networks need to be retrained to adapt the changes.

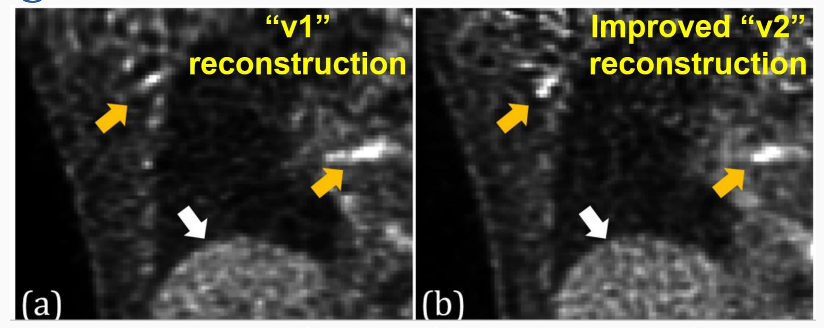


Fig: Change in image quality and appearance due to a change in reconstruction algorithm.

- A trained DNN often produces suboptimal predictions on unseen features.

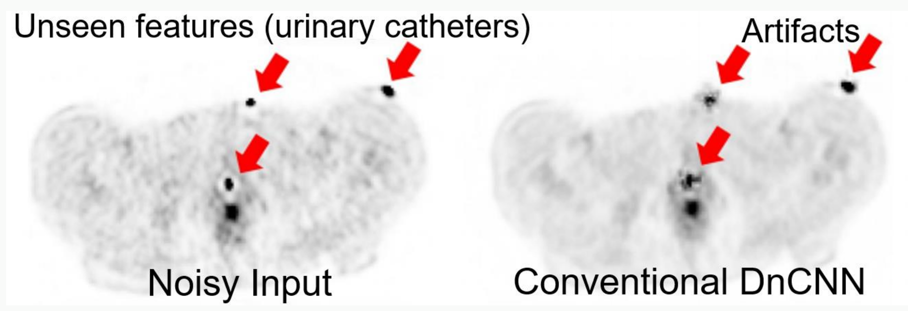
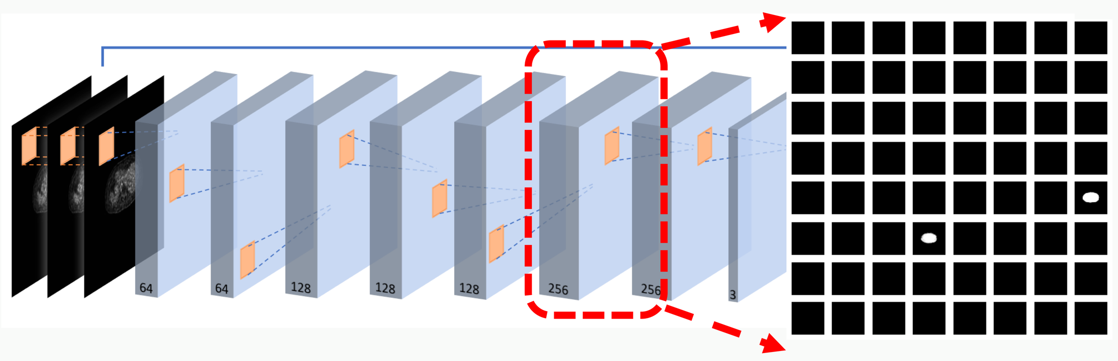


Fig: A denoising network produced artifacts on features that were not included in the training dataset.

In this study, we present Targeted Gradient Descent (TGD), a novel fine-tuning method that can extend a pre-trained network to a new task without revisiting data from the previous task while preserving the knowledge acquired from previous training. To a further extent, the proposed method also enables online learning of patient specific data. We demonstrate the proposed method's effectiveness in denoising tasks for PET images.

Rationale

- There are "Useless/redundant" feature maps exists in a ***pretrained*** ConvNet, because ConvNet did not efficiently use all of its kernels, and some of kernels contribute less.
- Can we specifically retrain these "useless" kernels that generates "useless/redundant" feature maps?



Method

- *Pretrained PET denoising ConvNet:*

A 2.5D DnCNN [1] that takes three consecutive 2D image slices as its input.

- *Identifying which feature maps are "meaningful".*

To update the specific kernels in the fine-tuning training, the information richness in the feature maps needs to be determined. The corresponding network kernels can then be identified and updated in the retraining stage to generate new feature maps. Here we used Kernel Sparsity and Entropy (KSE) metric proposed by Li et al. [2].

- *Kernel Sparsity and Entropy (KSE)*

KSE [] quantifies the sparsity and information richness in a kernel to evaluate a feature map's importance to the network. KSE contains two parts: the kernel sparsity, s_c, and the kernel entropy, e_c.

1. *Kernel sparsity s_c:* l1-norm of the kernels.
   - $s_c = \sum_{n=1}^{N} |W_{n,c}|$
2. *Kernel entropy e_c:* a measure of the diversity among the kernels.
   - $e_c = -\sum_{i=0}^{N-1} \frac{dm(W_{i,c})}{\sum_{i=0}^{N-1} dm(W_{i,c})} \log_2 \frac{dm(W_{i,c})}{\sum_{i=0}^{N-1} dm(W_{i,c})}$
3. *KSE score:*
   - $KSE = \sqrt{\frac{s_c}{1+\alpha \cdot e_c}}$

KSE is normalized to [0, 1] in each layer.

- *Targeted Gradient Descent (TGD)*

Identify the indices of the convolution kernels that generate the "useless" feature maps by setting a KSE threshold ϕ. The indices were used for generating a binary mask M_n in the gradient space:

- $M_n = \begin{cases} \mathbf{1}, & if\ KSE(Y_n) < \phi \\ \mathbf{0}, & if\ KSE(Y_n) \geq \phi \end{cases}$

M_n zeros out the gradients for the "useful" kernels (i.e., ones with $KSE(Y_n) \geq \phi$) during retraining (or fine-tuning). Mathematically, the back-propagation formula with TGD is defined as:

- $W_{n,c}^{(t+1)} = W_{n,c}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial Y_n^{(t)}} M_n X_c^{(t)} - \frac{\partial \mathcal{R}(W_{n,c}^{(t)})}{\partial Y_n^{(t)}} M_n X_c^{(t)}$

This masking process is packaged into a novel TGD layer that only activates during backpropagation and not forward pass.

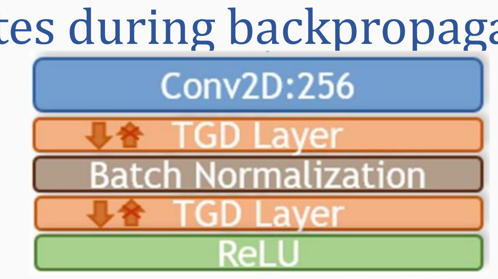


Fig: TGD layers are inserted after each convolution layer and batch normalization layer.

Method (cont.)

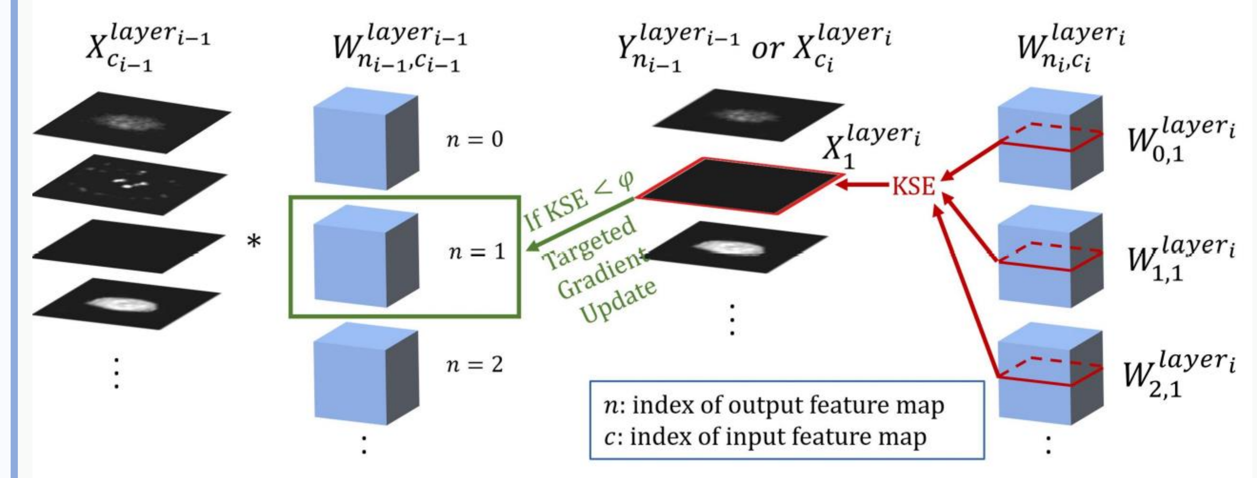


Fig: The framework of TGD training. The kernel weights in layer i (i.e., $W_{n_i,c_i}^{layer_i}$) were used to calculate KSE scores for the input feature maps in layer i (i.e., $X_{c_i}^{layer_i}$), then the kernels in layer $i-1$ (e.g., the green box: $W_{1,c_i}^{layer_i}$) that generated the input feature maps in layer i (i.e., $X_{c_i}^{layer_i}$) were identified and would be retrained in the proposed TGD method.

- *TGD noise-2-noise online learning*

Neural networks tend to produce suboptimal predictions on images that contain out-of-distribution features (features that are never seen in the training dataset). We then proposed to use TGD-network for N2N [9] online learning training, which alleviated hallucination artifacts from the images.

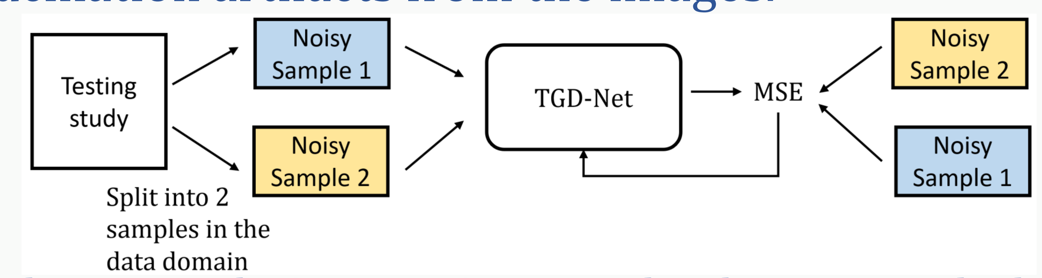


Fig: The proposed TGD noise-2-noise online learning method.

Experiment

We demonstrate the proposed TGD method on the task of PET image denoising.

- A DnCNN was trained using FDG PET images reconstructed from a prior version of the OSEM algorithm. We denote these images as v1 images and the pretrained DnCNN as the v1 network.
- The v1 network produces oversmoothed results when it is applied on the PET images reconstructed by an updated OSEM algorithm (we denote these images as v2 images).

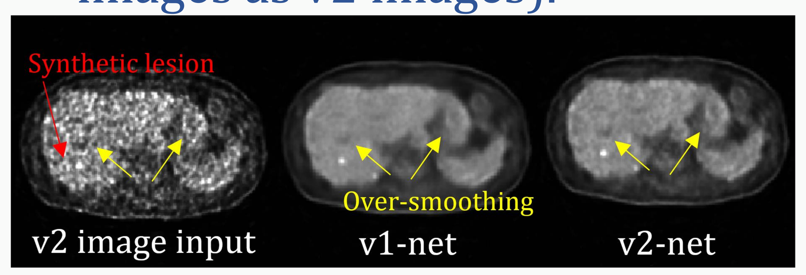


Fig: ConvNet denoised results of a v2 image generated by the v1 network and v2 network

The main goal is to use TGD fine-tuning to adapt the v1 network to v2 images, and then apply TGD N2N online learning to eliminate hallucination artifacts produced by out-of-distribution features.

Results

- *Compared methods*
- **Baseline networks:**
  - v1-net: DnCNN trained with v1 images
  - v2-net: DnCNN trained with v2 images
- **Fine-tuning task:**
  - FT-net: Fine-tuning the last three convolutional blocks.
  - TGD-net: v1-net fine-tuned using the TGD layers
- **Online-learning task:**
  - TGD_N2N-net: TGD N2N applied on the v2-net
  - TGD_N2N^2 -net : TGD N2N applied on the TGD-net
- *Determine KSE threshold ϕ*

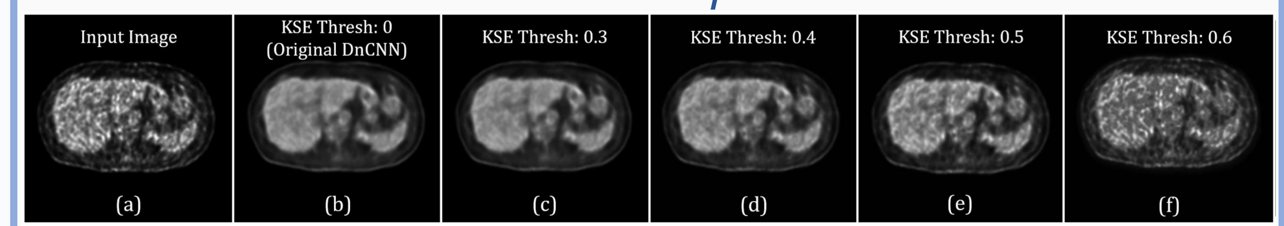


Fig: KSE threshold values of 0.3 and 0.4 resembles the original denoising performance the best.

- *TGD fine-tuning*

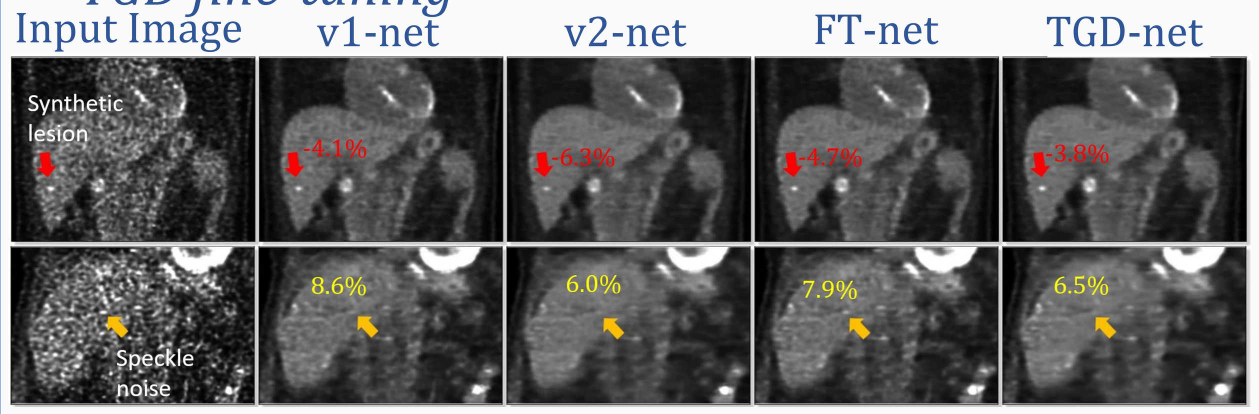


Fig: Qualitative comparisons between the proposed TGD method and other methods on denoising two FDG patient studies. The red numbers indicate the ensemble bias (%) comparing to the ground truth; the yellow numbers denote the liver CoV (%).

- *TGD N2N online-learning*

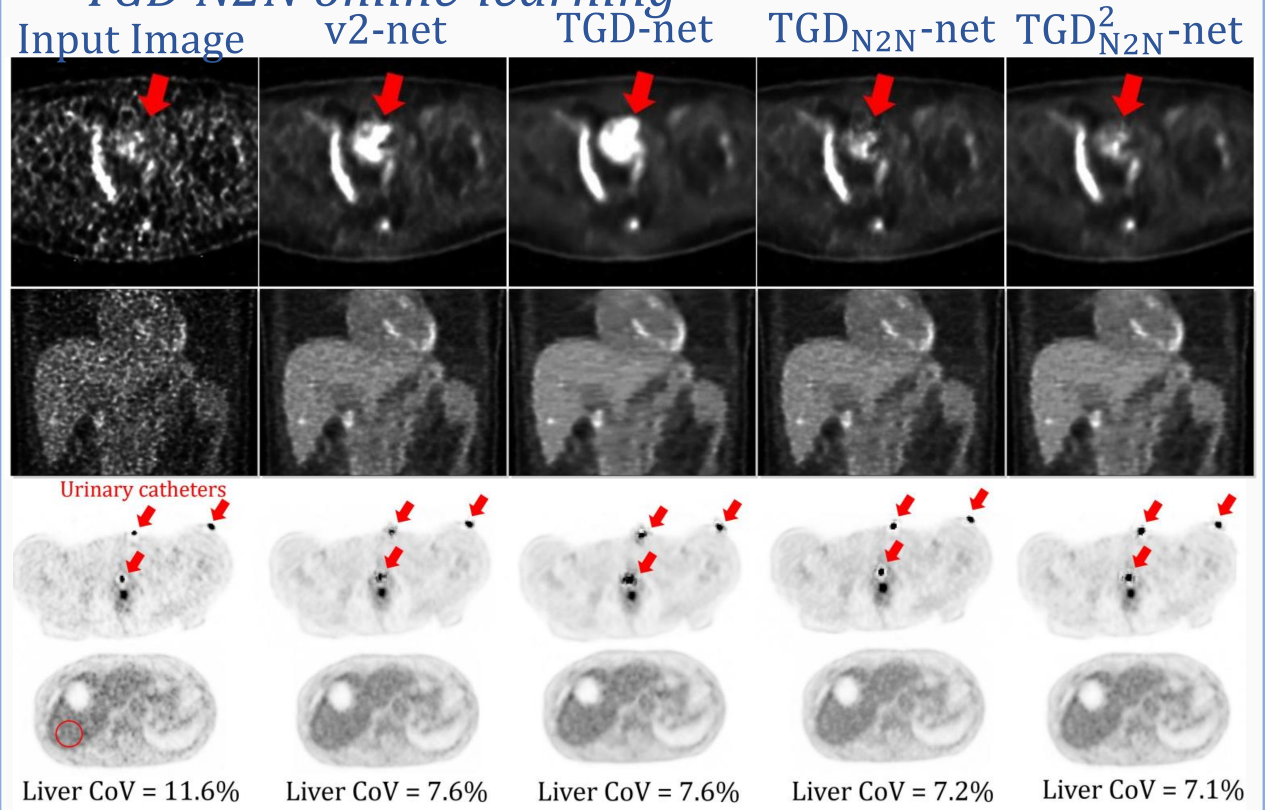


Fig: The red arrows indicate the unseen features, which was not included in any training datasets. The online learning approaches alleviated the artifacts while retaining similar denoising performance.

[1]: Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing 26(7), 3142–3155 (2017)

[2]: Li, Y., Lin, S., Zhang, B., Liu, J., Doermann, D., Wu, Y., Huang, F., Ji, R.: Exploiting kernel sparsity and entropy for interpretable cnn compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2800–28