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

Junyu Chen[1,2], Evren Asma[3], and 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
jchen245@jhmi.edu, {easma,cchan}@mru.medical.canon

Abstract. A convolutional neural network (ConvNet) is usually trained and then tested using images drawn from the same distribution. To generalize a ConvNet to various tasks often requires a complete training dataset that consists of images drawn from different tasks. In most scenarios, it is nearly impossible to collect every possible representative dataset as a priori. The new data may only become available after the ConvNet is deployed in clinical practice. ConvNet, however, may generate artifacts on out-of-distribution testing samples. 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. The method is built on the idea of reusing a pre-trained ConvNet's redundant kernels to learn new knowledge. We compare the performance of TGD to several commonly used training approaches on the task of Positron emission tomography (PET) image denoising. Results from clinical images show that TGD generated results on par with training-from-scratch while significantly reducing data preparation and network training time. More importantly, it enables online learning on the testing study to enhance the network's generalization capability in real-world applications.

Keywords: Fine-tuning · Online-learning · Image denoising.

1 Introduction

A Convolutional neural network (ConvNet) is usually trained and tested on datasets where images were sampled from the same distribution. However, if training and testing images are drawn from different distributions, ConvNets would suffer from performance degradation. This is a commonly observed scenario in medical imaging applications due to variations among patients, image acquisition, and reconstruction protocols [7, 8, 11]. For example, when applying denoising ConvNets on unseen features, it may cause artifacts in the denoised images as demonstrated in both Ultrasound [12] and Positron Emission Tomography (PET) applications [4]. To generalize a trained ConvNet to different image distributions, one has to include images sampled from the new distribution (task) in the training dataset and retrain the ConvNet. However, in medical

imaging, generating labeled datasets is often tedious, time-consuming, and expensive. In most scenarios, it is nearly impossible to collect every possible representative dataset as a priori. *The new data may only become available after the ConvNet is deployed.* Moreover, in product development, there is often a need to improve the imaging protocols (i.e., scan or reconstruction protocol) during a product development phase. The change of the image properties, i.e., local pixel correlations, would require regenerating all the training datasets with the updated protocol followed by retraining the denoising network. This recurring process is inevitably time and resource-consuming. Thus, it is more desirable to develop methods that can adapt to various image distributions with minimum need for additional training data and training time. Ultimately, the goal is to develop an online learning algorithm that can quickly retrain and adapt a pre-trained ConvNet to each testing dataset.

Fine-tuning is a promising approach to avoid training a ConvNet from scratch. During fine-tuning, a pre-trained network, usually trained using a large number of datasets from a different application, is used to continue the backpropagation on a smaller dataset from a new task [1, 9]. However, fine-tuning the network on a new task does not guarantee retaining the useful knowledge acquired from the previous training. If the number of training dataset from the new task is much less than that used in the old task, the fine-tuned network will overfit to the datasets in the new task with degraded generalization capability [10], which may not be suitable for the applications in which both tasks are of interest during testing. Another approach is using joint training (e.g., [2, 18]) or incremental learning (e.g., [3, 16, 17, 19]). They try to adapt a pre-trained network to new tasks while preserving the network's original capabilities. Joint training requires revisiting data from previous tasks while learning the new task [2, 15, 18] or modifying the network's architecture [16]. Incremental learning is used to continuously adapt a ConvNet to a constantly arriving data stream, enabling the ConvNet to learn new tasks incrementally and without forgetting the ones already learned. However, those methods typically require some level of revisiting training data from previous tasks and become unfeasible if the old data is unavailable.

Instead of blindly fine-tuning all the kernels in the specific layers or retraining the entire network with a mixture of new and old labels, it might be more sensible to precisely retrain the "meaningless" kernels to make them adapt to the new tasks while the "useful" kernels are preserved so they can retain the knowledge acquired from the prior training with a larger training dataset (a wider coverage of data distribution). This work proposes a novel fine-tuning method, the Targeted Gradient Descent (TGD) layer, that can be inserted into any ConvNet architecture. The novel contributions of the proposed
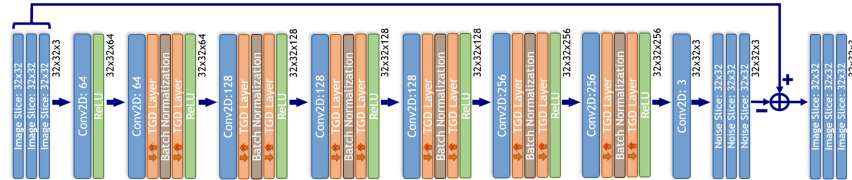

Fig. 1: The architecture of an eight-layer denoising residual network with proposed TGD layers (orange blocks).

method are 2-fold: 1. TGD 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; 2. It enables online learning that adapts a pre-trained network to each testing dataset to avoid generating artifacts on unseen features. We demonstrate the proposed method's effectiveness in denoising tasks for PET images.

2   Methodology

In this study, the pre-trained PET denoising ConvNet was built on the basis of the denoising convolutional neural network (DnCNN) [20]. The architecture of the DnCNN is the same as in Fig. 1 but *without* the TGD layers. It is a 2.5-dimensional network that takes three consecutive 2D image slices as its input. The network consists of eight 3×3 convolutional layers and a single residual layer at the end of the network. Each convolutional layer is followed by a batch normalization and a rectified linear unit (ReLU), except for the first and the last convolutional layers. The first convolution layer is followed by a ReLU, whereas the last convolution layer is not followed by any activation.

To update the specific kernels in the fine-tuning training, we first need to determine the information richness in the feature maps. The corresponding network kernels can then be identified and updated in the retraining stage to generate new feature maps, such that if the kernels can produce meaningful features, which are identified as "useful" kernels, while the kernels producing "meaningless" features are identified as "meaningless" kernels. However, it is hard to determine a feature map's information richness based solely on some particular input images because different input images may activate different feature maps. Here we used Kernel Sparsity and Entropy (KSE) metric proposed by Li et al. [14]. The KSE quantifies the sparsity and information richness in a kernel to evaluate a feature map's importance to the network. The KSE contains two parts: the kernel sparsity, s_c, and the kernel entropy, e_c, and they are briefly described here. We refer readers to [14] for details. The **kernel sparsity** for the cth input feature map is defined as:

$$s_c = \sum_{n=1}^{N} |W_{n,c}|, \tag{1}$$

where N denotes the total number output feature maps, $W_{n,c}$ denotes the 2D kernels, n and c are, respectively, the indices of the output and input feature maps. The **kernel entropy** is calculated as the entropy of the density metrics (i.e., $dm(\cdot)$):

$$e_c = -\sum_{i=1}^{N} \frac{dm(W_{i,c})}{\sum_{i=1}^{N} dm(W_{i,c})} \log_2 \frac{dm(W_{i,c})}{\sum_{i=1}^{N} dm(W_{i,c})}, \tag{2}$$

where $dm(W_{i,c}) = \sum_j A_{c_{ij}}$, and $A_{c_{ij}}$ is a nearest neighbor distance matrix for the cth convolutional kernel [14]. A small e_c indicates diverse convolution kernels. Thus, the corresponding input feature map provides more information to the ConvNet. KSE is then defined as:
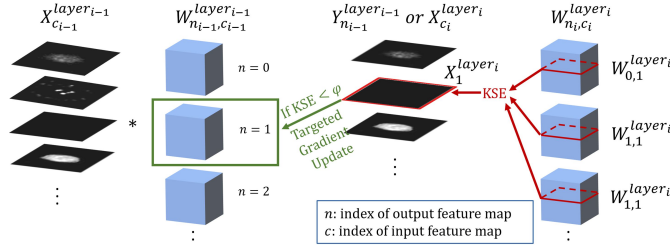
$$KSE = \sqrt{\frac{s_c}{1 + \alpha e_c}}, \tag{3}$$

Fig. 2: The framework of identifying kernels that generate redundant feature maps. The kernel weights in layer i (i.e., $W_{n,c}^{\text{layer}_i}$) were used to calculate KSE scores for the input feature maps in layer i (i.e., $X_c^{\text{layer}_i}$), then the kernels in layer i and $i-1$ (e.g., the green box: $W_{1,c_{i-1}}^{\text{layer}_{i-1}}$) that corresponded to calculated KSE scores for the input feature maps in layer i (i.e., X_c) were identified and would be retrained in the proposed TGD method.

where KSE, s_c, and e_c are normalized into [0, 1], and α is a parameter for controlling weight between s_c and e_c, which is set to 1 according to [14].

2.1   Targeted Gradient Descent Layer

The KSE score indicates the meaningfulness of feature maps to the ConvNet. Our goal is to retrain the kernels that generate redundant feature maps and keep the "useful" kernels unchanged. As illustrated in Fig. 2, we first calculate KSE for the input feature maps of layer i using the corresponding kernel weights from the ith convolutional layer. The feature maps with KSE scores below a certain user-defined threshold, ϕ, are marked as meaningless. We then identify and record the indices of the convolution kernels that generate the "meaningless" feature maps from the $(i-1)^{th}$ layer. The indices were used for creating a binary mask, M:

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

where ϕ is the user-defined KSE threshold. M_n zeros out the gradients for the "useful" kernels (i.e., $KSE(Y_n) \geq \phi$), so that these kernels will not be modified during retraining. The back-propagation formula is then adjusted to incorporate M_n as:

$$W_{n,c}^{(t+1)} = W_n^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial Y_n^{(t)}} M_n X_c^{(t)} - \frac{\partial \mathcal{R}(W_n^{(t)})}{\partial Y_n^{(t)}} M_n X_c^{(t)}. \tag{4}$$

We embedded the gradient zeroing process (i.e., $\frac{\partial \mathcal{L}}{\partial Y_n^{(t)}} M_n$) into a novel layer, named Targeted gradient descent layer (the orange blocks in Fig. 1). Notice that the batch normalization layers contain trainable weights as well (i.e., γ, β, and σ^2), where their gradients, $\frac{\partial \mathcal{L}}{\partial \gamma^{(t)}}$, $\frac{\partial \mathcal{L}}{\partial \beta^{(t)}}$, and $\frac{\partial \mathcal{L}}{\partial \sigma^{2(t)}}$ at iteration t can also be expressed as a function of $\frac{\partial \mathcal{L}}{\partial Y_n^{(t)}}$. As a result, the TGD layer was inserted after each convolutional layer as well as each batch normalization layer. Note that *the TGD layers are disabled during forward pass which means all kernels are activated.* During back-propagation, the TGD layers
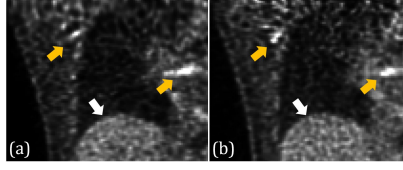
Fig. 3: The PET images of the right lung of a human subject reconstructed with (a) v1 reconstruction and (b) v2 reconstruction methods, respectively. The yellow arrows denote the structures that become elongated in v1 images due to the sub-optimal resolution modeling, which are corrected in the v2 reconstruction. The optimized resolution modeling also produced the liver noise texture (denoted by the white arrows) with a finer grain size (which represents a small correlation between neighboring pixels) in the (b) than (a).

are activated and only the targeted kernels are updated. The final architecture of the TGD-net is shown in Fig. 1.

The TGD-net adapted from a first task to the second task can further be adapted to a third task. The same TGD retraining process can be applied to the retrained TGD-net again, i.e., calculate KSE scores for the feature maps in the TGD-net, form new gradient masks, M_n, for the TGD layers, and then retrain the TGD-net with images from a third task. We name this recurring process TGD^n, where n represents the number of TGD retraining processes applied to a single network. In this work, we evaluated the cases in which $n = 1\&2$ (i.e., TGD and TGD^2).

3   TGD in PET Denoising

We demonstrate the proposed TGD method on the task of PET image denoising in two applications. We first use TGD to fine-tune an existing denoising ConvNet to make it adapt to a new reconstruction protocol using substantially fewer training studies. We then further use TGD in an *online-learning* approach to avoid the ConvNet generating artifacts (hallucinations) on unseen features during testing.

3.1   Fine-tuning

The pre-trained network was trained using FDG PET images acquired on a commercial SiPM PET/CT scanner reconstructed from a prior version of the OSEM algorithm. For simplicity, we denote these images as the v1 images and the denoising ConvNet trained using these images as the v1 network. We denote the PET images reconstructed by an updated OSEM algorithm as the v2 images and the corresponding denoising ConvNet as the v2 network. The system resolution modeling and scatter estimation in v2 reconstruction were optimized over the v1 reconstruction. Therefore, the noise texture in v2 images is finer, indicating a smaller correlation among neighboring pixels as shown in Fig. 3. The application of the v1 network on v2 images produced over-smoothed results and suppressed activity in small lesions, which could potentially lead to misdiagnosis.

Conventionally, whenever the reconstruction algorithm is updated, the entire training datasets have to be re-reconstructed, and the denoising network has to be retrained
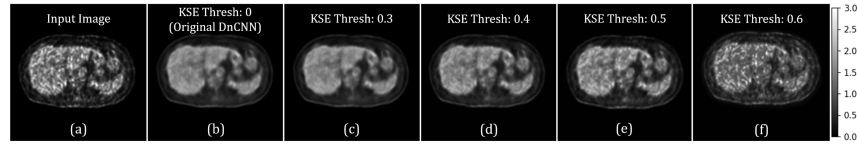
Fig. 4: The network's outputs produced by varying KSE threshold.

using the updated images for optimal performance, followed by qualitative and quantitative assessments on a cohort of testing studies. This process is extremely tedious and time-consuming.

The v1 network was trained using 20 x v1 whole-body FDG-PET human studies with a mixture of low (< 23) and high BMI (> 28). These studies were acquired for 10-min/bed, which were used as the target images. We uniformly subsampled the list mode data into 6 noise levels as 30, 45, 60, 90, 120, 180 sec/bed as the noisy inputs for noise adaptive training [6]. All these studies consist of 30,720 training slices in total. This v1 network was adapted using the TGD method to denoising v2 PET images. During the TGD's retraining stage, we used only 7 training datasets that consist of PET scans from patients with low BMI (< 23). However, the retrained network retained the knowledge on how to denoise PET scans of high BMI patients learned from the previous task (images of high BMI subjects are commonly substantially noisier than those of low BMI subjects). It is important to emphasize that the amount of v1 images used in v1 network training was significantly more than the amount of v2 images used in TGD fine-tuning. Based on this fact, we kept the weights of the noise classifier layer (i.e., the last convolutional layer) in the TGD-net unchanged during the retraining, thus avoiding the last layer from being biased by the v2 image data.

3.2   Online-learning

In the second experiment, we showed that TGD enables online-learning that further optimize the network's performance on each testing study and prevents artifacts (hallucination) from occurring on out-of-distribution features. This is achieved by using TGD with Noise-2-Noise (N2N) training scheme [5, 13]. Specifically, we rebinned a testing study list-mode data acquired with 120-sec into 2 noise realizations with equal count levels (60-sec). We used TGD to fine-tune the denoising network by using noise samples 1 and 2 as the inputs and noise samples 2 and 1 as the targets. We denote the online-learning network as TGD_{N2N}-net. To a greater extent, this procedure was also applied to the TGD-net from the first experiment (the network was TGD fine-tuned twice), and we denote the resulting network as TGD_{N2N}^2-net for convenience.

4   Experiments

The optimal KSE threshold was first studied. During the prediction, we dropped the kernels (i.e., setting the weights to zeros) identified as "meaningless" by the KSE threshold to examine whether these kernels indeed contributed less to the network. As shown in Fig. 4, where (a) shows an example slice of a PET scan, and (b) shows the denoised

	v1-net	v2-net	FT-net	TGD-net$^{0.3}$
Lesion Bias (%)	-6.30	-4.07	-4.71	**-3.77**
Liver CoV (%)	6.02	8.56	7.87	6.46

Table 1: Ensemble bias and CoV comparison results of the different training methods on the PET scan of the low-BMI patient.

PET image from the v1 DnCNN (this can be interpreted as having a KSE threshold = 0, because no kernel was dropped). We then arbitrarily tested four thresholds: 0.3, 0.4, 0.5, and 0.6. The larger the thresholds, the more the kernels were dropped. The percentage of the parameters that were dropped using the four thresholds are, respectively, 51.4%, 51.6%. 53.0%, and 67.3%, and the corresponding denoised results are shown in (c), (d), (e), and (f) of Fig. 4, respectively. The result from $\phi = 0.3$ is almost identical to the original DnCNN's result. Whereas, when $\phi > 0.4$, some severe artifacts begin to occur in the resulting images. Therefore, the KSE threshold, ϕ, was set to be 0.3 in this work.

TGD-net was compared to several baseline methods, including: 1. v1-net: A DnCNN trained using $20\times$ v1 PET images; 2. v2-net: A DnCNN trained using the same 20 studies but reconstructed with v2 algorithm; 3. FT-net: Fine-tuning the last three convolutional blocks [9] of v1-net using only $7\times$ v2 images; 4. TGD-net: v1-net fine-tuned using the TGD layers with $7\times$ v2 images (same studies as used in the FT-net). All networks were trained with 500 epochs.

The proposed $\mathrm{TGD}^2_{\mathrm{N2N}}$-net$^{\phi=0.3,0.4}$ and $\mathrm{TGD}_{\mathrm{N2N}}$-net$^{\phi=0.4}$ were built based on the previous TGD-net and v2-net, respectively. These networks were retrained using two noise realizations from a single study (i.e., N2N training). They were compared to: 1. v2-net: Same as above. 2. TGD-net$^{\phi=0.3}$: The TGD-net obtained from the previous task. The $\mathrm{TGD}_{\mathrm{N2N}}$ models were trained with 150 epochs.

All these methods were compared in terms of denoising on 2 FDG patient studies reconstructed with v2 algorithm (v2 images). One of the studies was acquired with 600-sec/bed with a simulated tumor that was inserted in the liver. We rebinned the list-mode study into 10×60-sec/bed image i.i.d noise realizations to assess the ensemble bias on the tumor and liver coefficient of variation (CoV) by using the 600-sec/bed image as the ground truth. We then further evaluated the methods on a second 60-sec/bed study.

4.1   Evaluation of TGD on Whole-Body PET Denoising

Fig. 5 shows the denoised results of the example cropped slices of the v2 PET images, where the figures in the first column represent the input image. Qualitatively, v1-net (the third column of Fig. 5) over-smoothed the v2 image that led to piece-wise smoothness in the liver and reduced uptake in the synthetic lesion compared to the results from other methods. In contrast, the result from v2-net (the second column) exhibited a higher lesion contrast with more natural noise texture (fine grain size) in liver regions. The fine-tuned networks (FT-net) yielded good performances on denoising the low-BMI-patient PET scans (the top figure of the fourth column) with higher lesion contrast. However, the speckle noise (denoted by the yellow arrow) in the high-BMI-patient PET
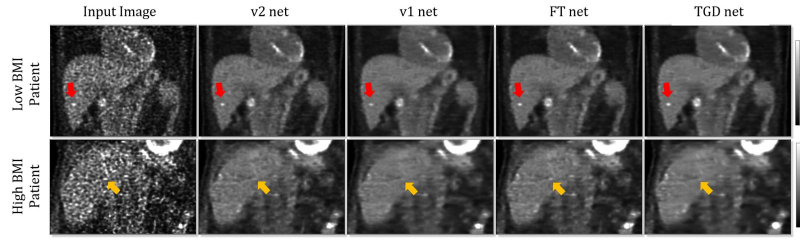
Fig. 5: Qualitative comparisons between the proposed TGD method and other methods on denoising two FDG patient studies. The red arrows indicate the synthetic lesion, and the yellow arrows highlight the difference.
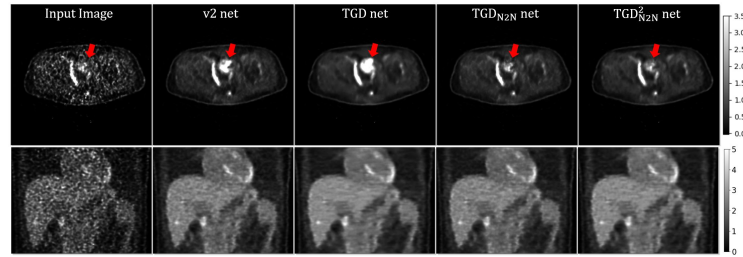

Fig. 6: Qualitative comparisons between the proposed TGD_{N2N} online learning and other methods. The red arrows indicate the artifactual feature generated by the v2-net and TGD-net around the bladder, which was not included in any training datasets. Both TGD_{N2N} and TGD_{N2N}^2 yielded images which are in high fidelity to the input image on the bladder, while retaining similar denoising performance as v2-net and TGD-net.

scans was also preserved. The proposed TGD-net yielded good lesion contrast but also low variations in the liver for both low- and high-BMI patient scans. The quantitative evaluations are shown in Table 1. The best performance is highlighted in bold. For the low-BMI patient study, the proposed method ($\phi = 0.3$) achieved the best lesion quantification with a small ensemble bias of -3.77% while maintaining a low-noise level of 6.45% in terms of CoV. In addition, fine-tuning a TGD net from the v1-net saved 64% of computational time compared to the training-from-scratch v2-net.

Fig. 5 shows the denoised results of the example cropped slices of the v2 PET images. Prior to TGD N2N training, v2-net and TGD-net created artifactual features (hallucination) around the bladder region (denoted by the red arrows). In contrast, the networks fine-tuned using the TGD N2N online learning scheme did not produce any artifacts, where the bladder's shape is nearly the same as that of the input image. To a greater extend, TGD_{N2N}^2-net$^{\phi=0.3,0.4}$ and TGD_{N2N}-net$^{\phi=0.4}$ retained the denoising performances of their base networks (i.e., v2-net and TGD-net).

5   Conclusion

This study introduced Targeted Gradient Descent, a novel incremental learning scheme that effectively reuses the redundant kernels in a pre-trained network. The proposed method can be easily inserted as a layer into an existing network and *does not require*

revisiting the data from the previous task. More importantly, it enables *online learning* on the testing study to enhance the network's generalization capability in real-world applications.

References

1. Amiri, M., Brooks, R., Rivaz, H.: Fine tuning u-net for ultrasound image segmentation: which layers? In: Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data, pp. 235–242. Springer (2019) 2
2. Caruana, R.: Multitask learning. Machine learning **28**(1), 41–75 (1997) 2
3. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European conference on computer vision (ECCV). pp. 233–248 (2018) 2
4. Chan, C., Yang, L., Asma, E.: Estimating ensemble bias using bayesian convolutional neural network. In: 2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). IEEE (2020) 1
5. Chan, C., Zhou, J., Yang, L., Qi, W., Asma, E.: Noise to noise ensemble learning for pet image denoising. In: 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). pp. 1–3. IEEE (2019) 6
6. Chan, C., Zhou, J., Yang, L., Qi, W., Kolthammer, J., Asma, E.: Noise adaptive deep convolutional neural network for whole-body pet denoising. In: 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC). pp. 1–4. IEEE (2018) 6
7. Chen, C., Dou, Q., Chen, H., Heng, P.A.: Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest X-ray segmentation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 11046 LNCS, pp. 143–151. Springer Verlag (sep 2018). https://doi.org/10.1007/978-3-030-00919-9_17 1
8. Ghafoorian, M., et al.: Transfer learning for domain adaptation in MRI: Application in brain lesion segmentation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 10435 LNCS, pp. 516–524. Springer Verlag (sep 2017). https://doi.org/10.1007/978-3-319-66179-7_59, https://ncigt.org/files/ncigt/files/ghafoorian-miccai2017.pdf 1
9. Gong, K., Guan, J., Liu, C.C., Qi, J.: Pet image denoising using a deep neural network through fine tuning. IEEE Transactions on Radiation and Plasma Medical Sciences **3**(2), 153–161 (2018) 2, 7
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 2
11. Kamnitsas, K., et al.: Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 10265 LNCS, pp. 597–609. Springer Verlag (2017). https://doi.org/10.1007/978-3-319-59050-9_47, https://link.springer.com/chapter/10.1007/978-3-319-59050-9{_}47 1
12. Laves, M.H., Tölle, M., Ortmaier, T.: Uncertainty estimation in medical image denoising with bayesian deep image prior. In: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis, pp. 81–96. Springer (2020) 1
13. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. In: ICML (2018) 6

14. 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. 2800–2809 (2019) 3, 4
15. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017) 2
16. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016) 2
17. Tasar, O., Tarabalka, Y., Alliez, P.: Incremental learning for semantic segmentation of large-scale remote sensing data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(9), 3524–3537 (2019) 2
18. Wu, C., Herranz, L., Liu, X., van de Weijer, J., Raducanu, B., et al.: Memory replay gans: Learning to generate new categories without forgetting. In: Advances in Neural Information Processing Systems. pp. 5962–5972 (2018) 2
19. Xiao, T., Zhang, J., Yang, K., Peng, Y., Zhang, Z.: Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 177–186 (2014) 2
20. 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) 3