Beware the Black-Box of Medical Image Generation: An Uncertainty Analysis by the Learned Feature Space

Yunni Qu[†], **David Yan**[‡], **Eric Xing**[§], **Fengbo Zheng**[¶], **Jie Zhang**[‡], **Liangliang Liu**[‖], **Gongbo Liang**[*],
[†] University of Toronto, Toronto, ON, Canada
[‡] University of Kentucky, Lexington, KY, USA
[§] Western Kentucky University, Bowling Green, KY, USA
[¶] Tianjin Normal University, Tianjin, China
[‖] Henan Agricultural University, Zhengzhou, Henan, China
[*] Texas A&M University-San Antonio, San Antonio, TX, USA

***Abstract*— Deep neural networks (DNNs) are the primary driving force for the current development of medical imaging analysis tools and often provide exciting performance on various tasks. However, such results are usually reported on the overall performance of DNNs, such as the Peak signal-to-noise ratio (PSNR) or mean square error (MSE) for imaging generation tasks. As a black-box, DNNs usually produce a relatively stable performance on the same task across multiple training trials, while the learned feature spaces could be significantly different. We believe additional insightful analysis, such as uncertainty analysis of the learned feature space, is equally important, if not more. Through this work, we evaluate the learned feature space of multiple U-Net architectures for image generation tasks using computational analysis and clustering analysis methods. We demonstrate that the learned feature spaces are easily separable between different training trials of the same architecture with the same hyperparameter setting, indicating the models using different criteria for the same tasks. This phenomenon naturally raises the question of which criteria are correct to use. Thus, our work suggests that assessments other than overall performance are needed before applying a DNN model to real-world practice.**

Index Terms — Neural Network, U-Net, Uncertainty

I. Introduction

Deep neural networks (DNNs) have shown promising performance on medical imaging-related tasks [1]–[6]. However, the evaluation of DNNs are often completed by reporting the overall performance, while insightful analysis is generally omitted. We believe analysis beyond the overall performance is needed and equally important, if not more, especially when applying neural networks to real-world tasks [7]–[9].

In recent years, DNN-based approach for medical image generation has been a rapidly developed field. The techniques are widely used for various tasks, such as image segmentation [10]–[12], image denoising [13], and image standardization [14]. As a data-driven approach, the features of DNNs are directly learned from the training set. The learning process is often guided by loss functions, such as mean square error (MSE) loss, for measuring the overall performance. This learning schema is constrained in finding the optimal overall solution for the given task. The feature space learning becomes a by-product of DNN training. Less constrains is applied to finding the optimal features.

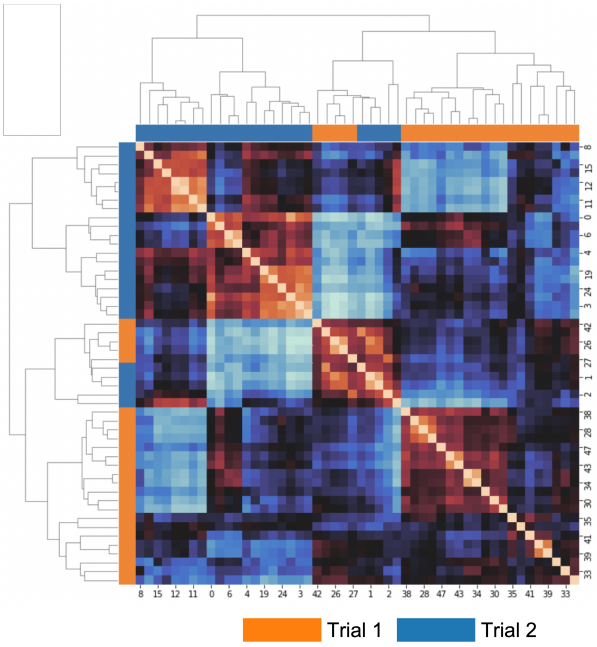


Fig. 1: Clustering heatmap shows the learned features of two training trials of the same U-Net are highly separable. The same samples are used to train and test the two training trials.

Figure 1 shows the clustering heatmap of features extracted by two U-Net [10] models. The two models are trained using the same architecture, same hyperparameters, and training data. Then, the same test samples are fed into the models for the feature extraction. Pair-wise correlation is applied to all the features extracted by the two models. Finally, unsupervised hierarchical clustering is applied to generate the figure. Each row (or column) shows the correlation of a given sample to all other samples. Rows and columns are the same. Color bars on the left and top indicate the specific model where the feature is extracted from (see Section III-B.2 for more details about the method). Ideally, numerous small clusters with two samples should be observed. Since same samples are fed into the two models,

the two features of a same sample should be naturally close to each other if the learned feature spaces of the models are similar. However, two large clusters are observed. Each of them contains samples from the one training trial, indicating the features are highly coupled with training trials. Such a phenomenon implies the output decisions of different models are based on different features, which is not rare to see in the deep learning world. However, it may not be acceptable in the medical domain since domain experts might be expected to make decisions based on the same standards, in general.

To better understand the inconsistency between the learned feature spaces, we train and evaluate twelve U-Net models, with three architectures and two training strategies, for abdominal CT images denoising. Our result suggests that though generative models, such as U-Net, may have a relatively stable overall performance across different training trails, the features used for decision making (i.e., image generation) are not consistent between training trials. This unaccepted behavior could lead to potential issues of medical applications. More assessments other than overall performance are needed before applying a DNN model to real-world practice. Guidelines and regulations are also needed to catch up with the AI advancement to ensure that models with claimed high overall performance undergo further assessment and validation before being applied to real-world practice.

II. Background

A. Neural Network Feature Space Learning

A typical neural network used in the imaging domain may be considered as the combination of numerous linear regression models and non-linear activation functions. The network is often denoted as $h_\Theta()$, where Θ is a list of tunable parameters or weights, $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$. Ideally, the feature space of a given task is embedded in Θ after the parameters are optimized. Loss functions, such as the MSE loss, are applied to guide neural network learnings by providing feedback during the training. The parameters, Θ, are optimized by minimizing the loss function. When a network is trained like this, the learning process is constrained in the overall performance and less constrained in the feature space.

B. Neural Network Weights Initialization

The goal of neural network training is to find the optimum weights for the model that output the desired result on a given task. The weights are iteratively updated during the training from an initial set. Undesired initial weights may lead to vanishing or exploding gradients issues [15], [16]. The most widely used method is to initialize the weights randomly, close to zero, and under the normal distribution.

The random weights are generated by a random number generator (RNG). Most RNGs used for programming are pseudorandom number generators (PRNGs) since generating truly random numbers is often non-trivial. PRNGs generate pseudorandom numbers using a pre-defined algorithm. A seed (a number or a vector) is needed to initialize a PRNG. With the same seed, the same sequence of pseudorandom numbers can be generated by the PRNG. Thus, when setting seed to a fixed number, the weights of a neural network are always initialized the same across multiple initializations.

III. Feature Space Evaluation Method

To understand the uncertainty of the learned feature space of image generating models, we selected three U-Net architectures—namely U-Net-5 with two encoding layers and two decoding layers, U-Net-7 with three encoding layers and three decoding layers, and U-Net-9 with four encoding layers and four decoding layers.

In addition to the network architecture, we also consider the effects of DNN weights initialization methods. Two seeding methods are used to initialize the models, random seeding and fixed seeding. Thus, six unique combinations (denoted as six unique models in the future) of architecture and seeding methods are considered in this work. Then, we train each model twice. The learned feature spaces between the two training trials of the same model are compared.

A. Feature Space Representation

We represent the learned feature space using the feature mapss (i.e., activations) extracted from the test images. We give the same test images to all the models and training trails. Then, we extract the activations of all the images from all the layers. For each model, two sets of $N \times M \times K$ feature maps are generated after this step, where N is the number of test images, $M \in [64, 512]$ is the number of feature maps generated by a particular layer, and $K \in [4, 8]$ is the number of layers of a specific model. The two sets of feature maps, finally, are compared using two computational-based methods and a clustering-based method.

B. Similarity Comparison

We evaluate the feature spaces consistency of a unique model by comparing the similarity of two learned feature spaces between the training trials using computational-based and clustering-based methods.

1) Computational-Based Methods: We use the cosine similarity (CS) and the Singular Vector Canonical Correlation Analysis (CCA) [17] as the computational evaluation metrics. Cosine similarity is widely used to measure the similarity between two non-zero vectors of an inner product space. The value is bounded in [0, 1]. Normally, the value 1 is considered as "identical", and 0 is considered as "not similar at all". We apply CS on the corresponding features of a test sample and report the mean CS across all the test samples.

Singular Vector Canonical Correlation Analysis is a tool for quickly comparing two representations in a way that is both invariant to affine transform. The original paper applies CCA to measure the similarity between layers of neural network models [17]. We use CCA to compare the learned feature spaces of two training trials that represented the features of each layer. We store the features of one layer of one training trial in a $N \times M$ matrix, where N is the number of samples and M is the number of features. Then, a single CCA value is calculated between two metrics that

is bounded in [0, 1], with a higher value indicating a higher degree of similarity.

2) Clustering-Based Method: In addition to the computational results, we also want to see whether the learned feature spaces of two training trails are separable. We apply hierarchical clustering on a set of features of two corresponding training trials. One particular pattern we are interested in is whether the clusters are coupled with the training trials.

Given N test samples, we extract the M features of one layer from each of the two training trials, $2N \times M$ features are generated after this step. Then, we compute the pair-wise correlation of the $2N \times M$ features and apply hierarchical clustering. If the learned feature spaces are quite different, we may see multiple clusters containing a large amount of samples from one training trail. Otherwise, no such patterns should be observed. Instead, we may see up to N small clusters, each only containing two samples. Since we test each sample twice, if the learned feature spaces are similar between the two training trials, the features of the same sample generated by two training trials should be closer to each other and be clustered together.

We present the result using a clustering heatmap. Each row or column is a data sample. The rows and columns are the same. The diagonal line should always have the highest correlation (i.e., 1) since it indicates the result compared with the sample itself. The color of the heatmap represents the pairwise correlation of each testing sample. The absolute value of the correlation is not critical for this analysis. We are more interested in the relative patterns (i.e., where the samples are coming from for each cluster). The color bar on the left and top indicate the training trail where the sample is coming from.

Figure 1 shows an example of the clustering heatmap that the learned feature spaces of the two training trials are highly separable. Two larger clusters are observed, with one containing features mainly generated by Training Trail 2 (samples are color-coded as blue), and the other containing features mainly generated by Training Trail 1 (samples are color-coded as orange).

IV. Experimental Result

A. Dataset

The abdominal CT image set of the 2016 Low Dose CT Grand Challenge [18] is used in this study. The image set contains both low dose and full-dose CT images for 50 patients. In total, 14760 abdominal CT slides were used in this study. We split the set to training/validation/test set on the patient level, with 9130 images in the training set, 3026 images in the validation set, and 2604 images in the test set. See Figure 2 for an example of the dataset.

B. Model Training

To evaluate the inconsistency of the learned feature space of image generating models, we trained six unique U-Net models with the combination of three architectures and two seeding methods, random seeding and fixed seeding.

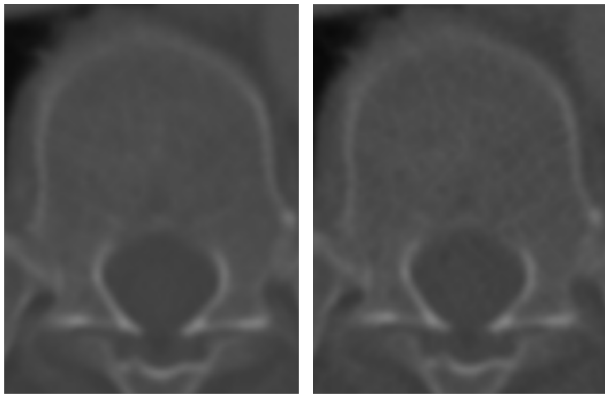
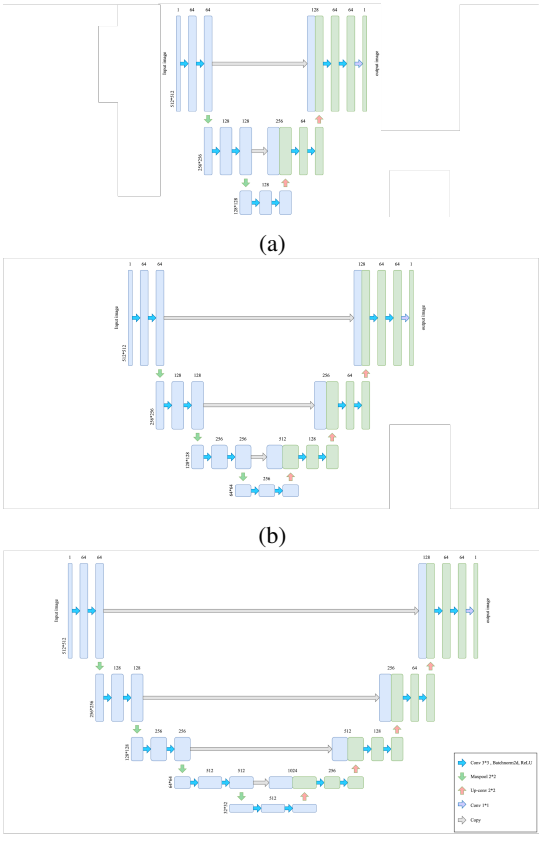


Fig. 2: Example of full dose (left) and low dose (right) CT images. The low dose CT image is nosier than the other one.



(a)

(b)

(c)

Fig. 3: U-Net architectures used in this study. (a) 5-layer architecture. (b) 7-layer architecture. (c) 9-layer architecture.

Figure 3 illustrates the three architectures. The contracting path consists of multiple implementations of double-convolution blocks (i.e., blocks with two convolution layers) with a kernel size of 3×3, each followed by a Batch Normalization and a Rectified Linear Unit, and then a 2×2 max pooling operation. The expansive path has a similar structure, except it applies 2×2 up-convolutions instead of the 2×2 max pooling operations at the down-sampling steps.

Each model was trained twice for 200 epochs with a batch

TABLE I: Mean Performance (PSNR) and Difference of the Two Trails of Each Model

Models	Mean PSNR (Difference)	
	Random Seeding	Fixed Seeding
U-Net-5	38.68 (1.85)	35.96 (5.59)
U-Net-7	40.33 (0.97)	40.14 (5.12)
U-Net-9	41.26 (1.16)	40.65 (0.50)

TABLE II: Feature Space CCA Analysis Difference Between Two Trials of Each Training Method

Layer	CCA (Larger is Better)					
	U5-R	U5-NR	U7-R	U7-NR	U9-R	U9-NR
Down1	0.0706	0.0943	0.0766	0.0894	0.0976	0.0798
Down2	0.0456	0.0635	0.1299	0.0583	0.0974	0.0755
Down3	–	–	0.1841	0.0538	0.1584	0.0697
Down4	–	–	–	–	0.1208	0.1052
Bottleneck						
Up1	0.1191	0.0599	0.1515	0.0534	0.1235	0.0754
Up2	0.1239	0.1727	0.1160	0.0446	0.0647	0.0681
Up3	–	–	0.1280	0.1138	0.1105	0.0571
Up4	–	–	–	–	0.1227	0.1103
Average	0.0898	0.0974	0.1310	0.0599	0.1794	0.0801

Ux: U-Net with x layers, -R: Random Seeding, -NR: Fixed Seeding

TABLE III: Feature Space Cosine Similarity Difference Between Two Trials of Each Training Method

Layer	Cosine Similarity (Larger is Better)					
	U5-R	U5-NR	U7-R	U7-NR	U9-R	U9-NR
Down1	0.6371	0.5726	0.5708	0.5670	0.5511	0.5506
Down2	0.6876	0.6141	0.5950	0.7157	0.6031	0.6510
Down3	–	–	0.5882	0.7257	0.6016	0.7220
Down4	–	–	–	–	0.6258	0.7039
Bottleneck						
Up1	0.6443	0.7120	0.6900	0.7076	0.5939	0.7519
Up2	0.6335	0.5392	0.7138	0.7571	0.7014	0.7058
Up3	–	–	0.6614	0.6364	0.7133	0.7562
Up4	–	–	–	–	0.6665	0.6375
Average	0.6479	0.6095	0.6366	0.6846	0.6332	0.6849

Ux: U-Net with x layers, -R: Random Seeding, -NR: Fixed Seeding

size of 8 and an RMSprop optimizer [19] initialized with a learning rate of 0.01. The combination of L1 (MAE) and L2 (MSE) were used as the loss function.

C. Result

Table I shows the denoising result of the six models. Each model is trained twice. The mean performances of each two training trials and the differences between the two training trials are reported. We use the PSNR to evaluate the quality of the generated images. Typical values for the PSNR in an 8-bit image are between 30 and 50 dB, where higher is better. The table shows that when increasing the number of layers of a U-Net model, the overall performance may be increased. The U-Net-9 model with random seeding generates the highest performance. The same architecture with fixed seeding achieved the second-best result. In general, random seeding models may have slightly higher performance than fixed seeding models. In addition, the performance of random seeding models may be more stable than the fixed seeding models across multiple training.

Tables II and III show the feature space analysis result using CCA and cosine similarity (CS), respectively. The tables reveal that both the CCA and CS are relatively low for all the models. This result may imply that none of the models is likely to have a consistent learned feature space across multiple training trials. However, we observed one interesting pattern that is the decoder part (i.e., the up layers) often has higher values than the encoder part (i.e., the down layers). Such an observation indicates the feature spaces of the decoder part may be more consistent than the encoder part of a given U-Net model.

Figure 4 shows the clustering heatmap of U-Net-9 model with fixed seeding. The figure reveals that the learned feature spaces of the models across all the layers are highly spreadable by unsupervised clustering algorithms. Large clusters containing samples from only one training trail are observed in all figures. Similar results are observed for all other models. The clustering-based evaluation also indicates that none of the models is likely to have a consistent learned feature space across multiple training trials.

V. Discussion

As a data-driven approach, a neural network learns features directly from the training set. Due to the exciting overall performance of neural networks, such features are often thought of as more robust or directly related to the predictive task than conventional hand-crafted features. Researchers are also seeking to find hidden patterns from the deep learning features that can be used to understand a given task [20]–[22]. However, the learning of a neural network model is often guided by loss functions measuring the overall performance. This learning schema is less constrained in the feature space learning than the overall performance that may provide an open ending.

This work evaluates the feature space consistency of six unique U-Net models. According to the computational-based and clustering-based evaluation results, none of the models are likely to have a consistent learned feature space across multiple training trials. Surprisingly, fixed seeding does not lead to a stable feature space either. This is contradicted to our intuition. Since all the training trials are initialized with the same set of weights for fixed seeding, we would think it might lead to a more stable feature space learning. However, no such behavior is observed in our study.

In addition, our result shows that the degree of feature space inconsistency might not be coupled with performance differences. For instance, we would naturally assume that the larger inconsistency between the learned feature spaces may lead to larger performance differences. However, our result shows that this assumption is not true. For instance, the U-Net-9 models with random seeding have a mean PSNR of 41.26 and the PSNR difference between the two training trials is 1.16. The U-Net-9 with fixed seeding has a mean PSNR of 40.65 with a difference of 0.50 difference between the two training trials. However, the CCA analysis shows the feature space of the latter one is less stable.

The results carried out through this study suggest only evaluating the overall performance, such as MSE or PSNR,

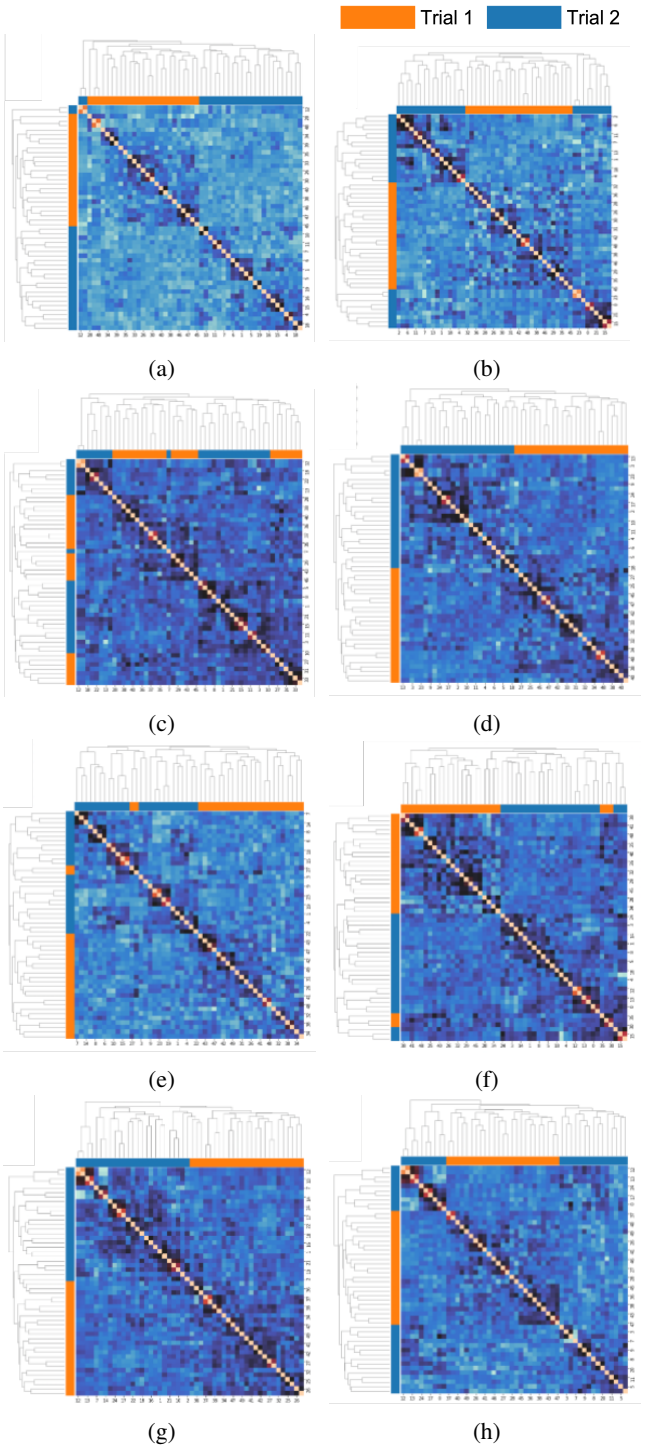


Fig. 4: Clustering heatmap the feature space similarity of each layer in the U-Net-9 model with fixed seeding. (a)-(d): Down1 to Down4 layers. (e)-(h): Up1 to Up4 layers.

is not sufficient enough to provide a complete understanding of a neural network model. Novel training schema may be needed to provide a more stable feature space across multiple training trials.

REFERENCES

[1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[2] Y. Zhang *et al.*, "2d convolutional neural networks for 3d digital breast tomosynthesis classification," in *IEEE International Conference on Bioinformatics and Biomedicine*, 2019.

[3] G. Liang, X. Wang, Y. Zhang, and N. Jacobs, "Weakly-supervised self-training for breast cancer localization," in *Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2020.

[4] X. Xing *et al.*, "Dynamic image for 3d mri image alzheimer's disease classification," in *European Conference on Computer Vision*. Springer, 2020, pp. 355–364.

[5] G. Liang *et al.*, "Contrastive cross-modal pre-training: A general strategy for small sample medical imaging," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2021.

[6] Q. Ying *et al.*, "Multi-modal data analysis for alzheimer's disease diagnosis: An ensemble model using imagery and genetic features," in *43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2021.

[7] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, "Calibrating predictive model estimates to support personalized medicine," *Journal of the American Medical Informatics Association*, vol. 19, no. 2, pp. 263–274, 2012.

[8] G. Liang, Y. Zhang, X. Wang, and N. Jacobs, "Improved trainable calibration method for neural networks on medical imaging classification," in *British Machine Vision Conference (BMVC)*, 2020.

[9] X. Wang, G. Liang, Y. Zhang, H. Blanton, Z. Bessinger, and N. Jacobs, "Inconsistent performance of deep learning models on mammogram classification," *Journal of the American College of Radiology*, vol. 17, no. 6, pp. 796–803, 2020.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

[11] R. P. Mihail, G. Liang, and N. Jacobs, "Automatic hand skeletal shape estimation from radiographs," *IEEE transactions on nanobioscience*, vol. 18, no. 3, pp. 296–305, 2019.

[12] L. Liu, J. Chang, Y. Wang, P. Zhang, G. Liang, and H. Zhang, "Ll-rhnet: Multiple lesions segmentation using local-long rang features," *Frontiers in Neuroinformatics*, 2022.

[13] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1348–1357, 2018.

[14] G. Liang, S. Fouladvand, J. Zhang, M. A. Brooks, N. Jacobs, and J. Chen, "Ganai: Standardizing ct images using generative adversarial network with alternative improvement," in *IEEE International Conference on Healthcare Informatics*, 2019.

[15] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[16] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[17] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, 2017.

[18] C. H. McCollough *et al.*, "Low-dose ct for the detection and classification of metastatic liver lesions: results of the 2016 low dose ct grand challenge," *Medical physics*, vol. 44, no. 10, pp. e339–e352, 2017.

[19] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

[20] Z. A. Shboul, M. Alam, L. Vidyaratne, L. Pei, M. I. Elbakary, and K. M. Iftekharuddin, "Feature-guided deep radiomics for glioblastoma patient survival prediction," *Frontiers in neuroscience*, vol. 13, p. 966, 2019.

[21] "Integrating deep and radiomics features in cancer bioimaging."

[22] K. Kobayashi, M. Miyake, M. Takahashi, and R. Hamamoto, "Observing deep radiomics for the classification of glioma grades," *Scientific Reports*, vol. 11, no. 1, pp. 1–13, 2021.