Journal of Medical Imaging

MedicalImaging.SPIEDigitalLibrary.org

Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks

Shuyue Guan
Murray Loew

SPIE.

Shuyue Guan, Murray Loew, "Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks," *J. Med. Imag.* **6**(3), 031411 (2019), doi: 10.1117/1.JMI.6.3.031411.

Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks

Shuyue Guan and Murray Loew*
George Washington University, Medical Imaging and Image Analysis Laboratory, Department of Biomedical Engineering, Washington DC, United States

Abstract. The convolutional neural network (CNN) is a promising technique to detect breast cancer based on mammograms. Training the CNN from scratch, however, requires a large amount of labeled data. Such a requirement usually is infeasible for some kinds of medical image data such as mammographic tumor images. Because improvement of the performance of a CNN classifier requires more training data, the creation of new training images, image augmentation, is one solution to this problem. We applied the generative adversarial network (GAN) to generate synthetic mammographic images from the digital database for screening mammography (DDSM). From the DDSM, we cropped two sets of regions of interest (ROIs) from the images: normal and abnormal (cancer/tumor). Those ROIs were used to train the GAN, and the GAN then generated synthetic images. For comparison with the affine transformation augmentation methods, such as rotation, shifting, scaling, etc., we used six groups of ROIs [three simple groups: affine augmented, GAN synthetic, real (original), and three mixture groups of any two of the three simple groups] for each to train a CNN classifier from scratch. And, we used real ROIs that were not used in training to validate classification outcomes. Our results show that, to classify the normal ROIs and abnormal ROIs from DDSM, adding GAN-generated ROIs in the training data can help the classifier prevent overfitting, and on validation accuracy, the GAN performs about 3.6% better than affine transformations for image augmentation. Therefore, GAN could be an ideal augmentation approach. The images augmented by GAN or affine transformation cannot substitute for real images to train CNN classifiers because the absence of real images in the training set will cause over-fitting. *© 2019 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JMI.6.3.031411]

Keywords: breast mass classification; deep learning; convolutional neural networks; generative adversarial networks; image augmentation; image synthesis; mammogram; computer-aided diagnosis.

Paper 18219SSR received Oct. 4, 2018; accepted for publication Feb. 22, 2019; published online Mar. 23, 2019.

1 Introduction

Breast cancer is the second leading cause of death among US women and will be diagnosed in about 12% of them.[1,2] The commonly used mammographic detection based on computer-aided detection (CAD) methods can improve treatment outcomes for breast cancer and increase survival times.[3] These traditional CAD tools, however, have a variety of drawbacks because they rely on manually designed features. The process of handcrafted feature design can be tedious, difficult, and nongeneralizable.[4] In recent years, developments in machine learning have provided alternative methods to CAD for feature extraction; one is to learn features from whole images directly through a convolutional neural network (CNN).[5,6] Usually, training the CNN from scratch requires a large number of labeled images;[7] for example, the AlexNet (a classical CNN model) was trained by using about 1.2 million labeled images.[8] For some kinds of medical image data, such as mammographic tumor images, it is difficult to obtain a sufficient number of images to train a CNN classifier because the true positives are scarce in the datasets and expert labeling is expensive.[9] The shortcomings of having an insufficient number of images to train a classifier are well known,[8,10] so it is worthwhile to examine image augmentation as a way to create new training images and thus to improve the performance of a CNN classifier.

Previous approaches to image augmentation used original images modified by rotation, shifting, scaling, shearing, and/or flipping. We name the original images ORG images, and the images augmented by affine transformation AFF images in the rest of this paper. The potential problem with such processing is that slightly changed images are similar to original ones; they may not be used as new training images to improve the performance of a CNN classifier. Large changes, on the other hand, may change the structure or pattern of objects in training images and degrade the performance of the classifier. An alternative image augmentation method is to generate synthetic images using the features extracted from original images. These generated images are not exactly like the original ones but could keep the essential features, structures, or patterns of the objects in original images. For this purpose, the generative adversarial network (GAN) is a good candidate for augmenting the training dataset. As with CNN, GAN is a neural network-based learning method introduced by Goodfellow et al.,[11] and it is a state-of-the-art technique in the field of deep learning.[12] GAN has many applications in the field of image processing, for example, image translation,[13,14] object detection,[15] super-resolution,[16] and image blending.[17] Recently, various GANs are also developed for the medical imaging, such as GANCS[18] for MRI reconstruction, SegAN,[19] DI2IN,[20] and SCAN[21] for medical

*Address all correspondence to Murray Loew, E-mail: loew@gwu.edu

2329-4302/2019/$25.00 © 2019 SPIE

image segmentation. In our previous work,[22] GAN images are the augmented images generated from GAN.

To compare the performances of GAN images with AFF images for image augmentation, we first cropped the regions of interest (ROIs) from images in the digital database for screening mammography (DDSM)[23] database as the original (ORG) ROIs. Second, by using these ORG ROIs, we applied GAN to generate the same number of GAN ROIs. We also used ORG ROIs to generate the same number of AFF ROIs. Then, we used six groups of ROIs: GAN ROIs, AFF ROIs, ORG ROIs, and three mixture groups of any two of the three simple ROIs to train a CNN classifier from scratch for each group. We used the remainder of the ORG ROIs (that were never used in augmentation and training) to validate classification outcomes. Our results demonstrate that to classify the normal ROIs and abnormal ROIs from DDSM, adding GAN ROIs to the training data can improve classification performance and the improvement is (about 3.6%) better than adding AFF ROIs. The maximum validation accuracy for training by only GAN ROIs is about 80%; it shows that the synthetic ROIs generated from a GAN can retain some important features, structure, or patterns from ORG ROIs. Since GAN performs better than affine transformation, GAN could be a good augmentation option.

2 Methods

2.1 *Mammogram Databases and Image Preprocessing*

Mammography is the process of using low-energy x-rays to examine the human breast for diagnosis and screening. There are two main orientations for acquisition of the x-ray images: the cranio-caudal (CC) view and the mediolateral-oblique (MLO) view (Fig. 1). The goal of mammography is the early detection of breast cancer,[24] typically through detection of masses or abnormal regions from the x-ray images. Usually, such abnormal regions are spotted by doctors or expert radiologists. In this study, we used mammograms from the DDSM.[23] It is a mammographic images resource used widely by researchers in mammographic image analysis. It is a collaborative effort between Massachusetts General Hospital, Sandia National Laboratories, and the University of South Florida Computer Science and Engineering Department. The DDSM database contains ~2620 mammograms in total: 695 normal mammograms, 1925 abnormal mammograms (914 malignant/cancers, 870 benign, and 141 benign without callback) with locations and boundaries of abnormalities. Each case includes four images representing the left and right breasts in CC and MLO views.

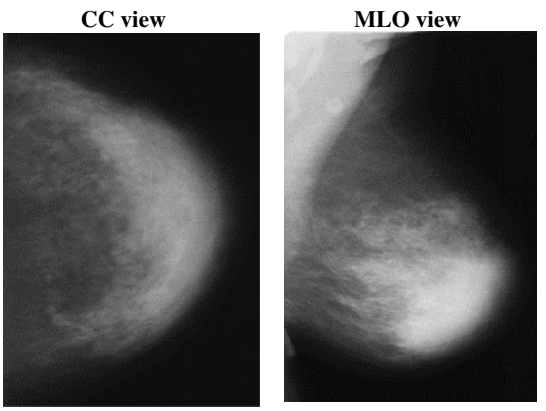

Fig. 1 Mammography in CC and MLO views.

We downloaded all mammographic images from DDSM's official website.[25] Images in DDSM are compressed in LJPEG format. To decompress and convert these images, we used the DDSM utility.[26] We converted all images in DDSM to PNG format. DDSM describes the location and boundary of actual abnormality by chain-codes, which are recorded in OVERLAY files for each breast image containing abnormalities. The DDSM utility also provides the tool to read boundary data and display them for each image having abnormalities. Since the DDSM utility tools run on MATLAB, we used it to implement all preprocessing tasks. We used the ROIs instead of entire images to train CNN classifiers. These ROIs are cropped rectangle-shape images and obtained by:

- For abnormal ROIs from images containing abnormalities, they are the minimum rectangle-shape areas surrounding the whole given ground-truth boundaries.
- Normal ROIs were cropped from the contralateral breast; the region was the same size and in the corresponding location as the tumor on the ipsilateral side. If both left and right breasts had abnormal ROIs and their locations overlapped, we discarded this sample. Since in most cases only one breast had a tumor, and the area and shape of the left and right breasts were similar, normal and abnormal ROIs had similar black background areas and scaling.

The selected ROIs for this work have no black background areas, the shapes are close to square (width-height ratio <1.2) and the sizes are larger than 320×320 pixels (to avoid upsampling). The sizes of abnormal ROIs vary with abnormality boundaries. Since the CNN requires all input images to be one specific size and the usual inputs for CNN are RGB images (images in DDSM are grayscale), we resized the ROIs by resampling and converted them to RGB (three-layer cubes) by duplication (Fig. 2). These images cropped from mammogram are ORG ROIs.

2.2 *Image Augmentation by Affine Transformation*

The image augmentation by affine transformations that we applied on ORG ROIs is: rotation, width shifting, height

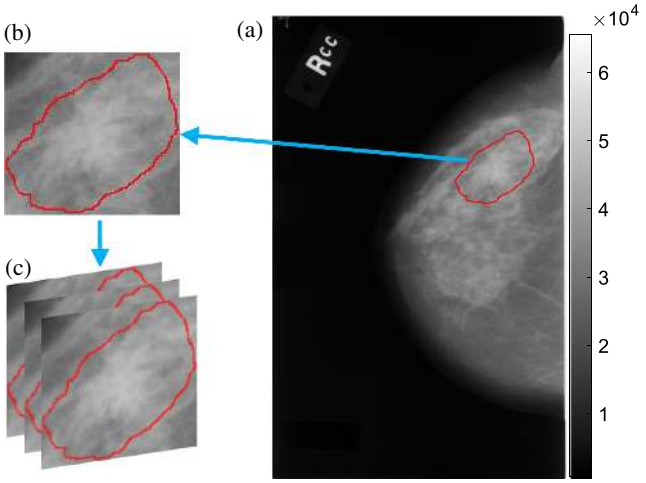

Fig. 2 (a) A mammographic image from DDSM rendered in grayscale; (b) cropped ROI by the given truth abnormality boundary; and (c) convert gray to RGB image by duplication.

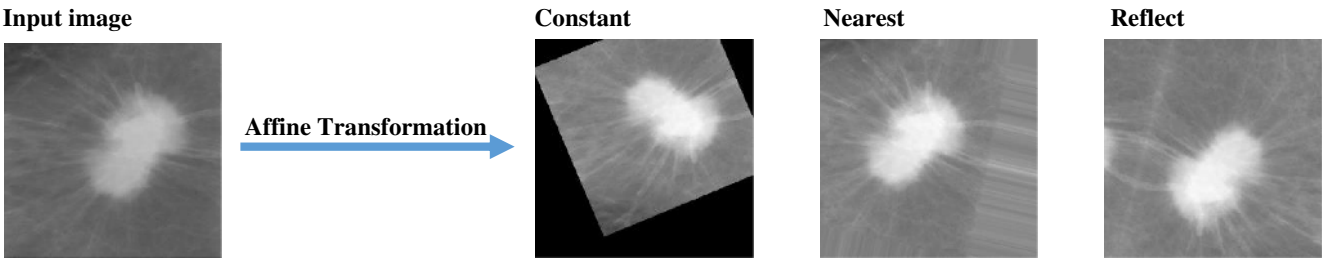

Fig. 3 The three affine transformations.

shifting, shearing, scaling, horizontal flipping, and vertical flipping. All transformations were applied randomly and some are in defined ranges. The range of rotation was 0 deg 30 deg and width shifting, height shifting, shearing, and scaling were 0% to 20% according to the total image size. Since the input image size and position must in general change after affine transformations, we used padding (filling) points outside the boundaries to maintain the size of the output image. There are three commonly used padding methods: set a constant value for all pixels outside the boundaries, copy the values at the nearest pixel on the boundaries, and reflect the image around the boundaries. Figure 3 shows the results of the three padding methods. We will choose to use the padding method that can obtain the best classification accuracy.

2.3 *Image Augmentation by GAN*

The GAN is a neural-network-based generative model that learns the probability distribution of real data and creates simulated data samples with a similar distribution (Fig. 4). Formally, in d-dimensional space, for $x \in R^d$, $y = p_{\text{data}}(x)$ is a mapping from x to real data y. We create a neural network called the generator G to simulate this mapping. If sample y comes from p_{data}, it is a real one; if sample z comes from G, it is a synthetic one. Another neural network, the discriminator D, is used to detect whether a sample is real or synthetic. Ideally, $D(y) = 1$; $D(z) = 0$. The two neural networks G and D compose the GAN. We can find G and D by solving the two-player minimax game,[11] with value function $V(G, D)$:

$$\min_G \max_D V(G,D) = \mathrm{E}\{\log\, D[p_{\text{data}}(x)]\} + \mathrm{E}(\log\{1 - D[G(x)]\}). \quad (1)$$

This min-max problem has a global optimum (Nash equilibrium) solution for $G(x) = p_{\text{data}}(x)$. That is the goal: to find the distribution of real data. At equilibrium, the discriminator D can no longer distinguish the real from the synthetic sample, where $D(y) = D(z) = 0.5$. Synthetic samples can be generated from G by changing the input x. In this study, the input x for G was a noise vector having 100 elements from a Gaussian distribution $\sim N(0,1)$. The key point of a well-trained GAN is that it can generate seemingly real data samples from noise vectors. To train a GAN, we used a limited number of real samples. Ideally, GAN could generate unlimited different synthetic samples.

To implement GAN, we built the generator and discriminator neural networks. The details about their structures are shown in Table 1. The generator consisted of four upsampling layers to double the size of the image and five convolutional layers. The activation function for each layer was the ReLU function[27] except the last one for output, which was a tanh function. The function of the generator is to transform a 100-length vector to a $320 \times 320 \times 3$ image. The input of the discriminator is a $320 \times 320 \times 3$ image and its output is a value between 0 and 1,

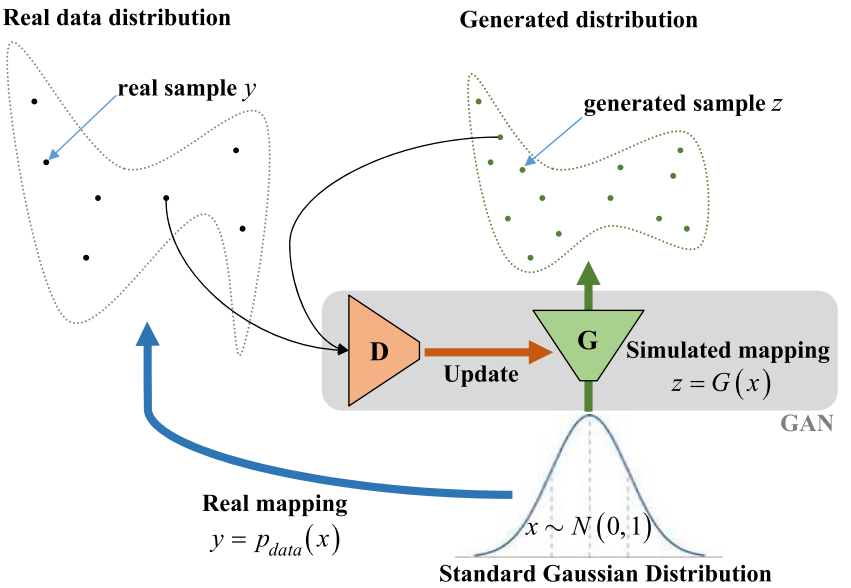

Fig. 4 The principle of GAN.

Table 1 The architecture of generator and discriminator neural networks.

Layer	Shape
Generator	
Input: 100-length vector	100
FC_($256 \times 20 \times 20$) + ReLU	102400
Reshape to $20 \times 20 \times 256$	$20 \times 20 \times 256$
Normalization + Up-sampling	$40 \times 40 \times 256$
Conv_3-256 + ReLU	$40 \times 40 \times 256$
Normalization + Up-sampling	$80 \times 80 \times 256$
Conv_3-128 + ReLU	$80 \times 80 \times 128$
Normalization + Up-sampling	$160 \times 160 \times 128$
Conv_3-64 + ReLU	$160 \times 160 \times 64$
Normalization + Up-sampling	$320 \times 320 \times 64$
Conv_3-32+ ReLU	$320 \times 320 \times 32$
Normalization + Conv_3-3+ ReLU	$320 \times 320 \times 3$
Output (tanh): $[-1, 1]$	$320 \times 320 \times 3$
Discriminator	
Input: RGB image	$320 \times 320 \times 3$
Conv_3-32 + ReLU	$320 \times 320 \times 32$
MaxPooling_2 + Dropout (0.25)	$160 \times 160 \times 32$
Conv_3-64 + ReLU	$160 \times 160 \times 64$
MaxPooling_2 + Dropout (0.25)	$80 \times 80 \times 64$
Conv_3-128 + ReLU	$80 \times 80 \times 128$
MaxPooling_2 + Dropout (0.25)	$40 \times 40 \times 128$
Conv_3-256 + ReLU	$40 \times 40 \times 256$
MaxPooling_2 + Dropout (0.25)	$20 \times 20 \times 256$
Flatten	102400
FC_1	1
Output (sigmoid): [0, 1]	1

where "0" indicates that D has decided that the image is synthetic, and "1" that the image is real. As with a typical CNN, the discriminator had four convolutional layers with max-pooling layers and one fully connected (FC) layer. The activation function for each convolutional layer was also the ReLU function and the last one for output was a sigmoid function, which mapped the output value to the range of [0, 1].

The notation Conv_3-32 means there are 32 convolutional neurons (units) and the filter size in each unit is 3×3-pixel (height × width) in this layer. MaxPool_2 means a max-pooling layer with the filters defined by a 2×2-pixel window, stride 2. FC_n means a fully connected layer having n units. The dropout layer[28] randomly set a fraction rate of input units to 0 for the next layer at every updating during training; it helped the networks avoid overfitting. Our training optimizer was Nadam[29] using default parameters (except the learning rate changed to 1e-4), the loss function was binary cross entropy, the updating metric was accuracy, the batch size was 30, and the number of total epochs was set to be 1e+5.

The training methods of GAN are:

- Step 1: Randomly initialize all weights for both networks.
- Step 2: Input a batch of length-100 noise vectors to generator to obtain synthetic images.
- Step 3: Train the discriminator by a batch of synthetic images labeled "0," and real images labeled "1".
- Step 4: To train the generator: input a batch of length-100 noise vectors to the generator to obtain synthetic images and label them as "1." Then, input these synthetic images to the discriminator to obtain the predicted labels. The differences between predicted labels and "1" will be the loss for updating the generator. It is noteworthy that in this step, only the weights in the generator were changed; weights in the discriminator were fixed.
- Step 5: Repeat step 2 to step 4 until all real images have been used once; that is one epoch. When the number of epochs reaches a certain value, training stops.

For the step 5, the ideal situation is to stop training when the classification accuracy of the discriminator converges to 50%. That means the discriminator no longer can distinguish the real images from the synthetic images generated from a well-trained generator. The discriminator plays a role as an assistant in GAN. After training, we used the generator neural networks to generate synthetic images.

2.4 *CNN for Classification*

A CNN was designed as the discriminator in GAN. Its function was to distinguish real and synthetic mammographic ROIs. We also built a CNN to classify abnormal ROIs and normal ROIs, and it was called CNN tumor classifier. As shown in Table 2, this CNN classifier consisted of three convolutional layers with max-pooling layers and two FC layers. The activation function for each layer was the ReLU function except the last one for output. The output layer used a sigmoid function, which mapped the output value to the range [0, 1]. Its input was an image of size 320×320 pixels. Since the sigmoid function was used in the output layer, the predicted outcome from the CNN classifier was a value between 0 and 1. By default, the classification threshold was 0.5, meaning that if the value was less than 0.5 it was considered as "0" (normal), otherwise it was considered as "1" (abnormal). The optimizer for training was Nadam using default parameters[30] (except the learning rate was changed to 1e-4), the loss function was binary cross entropy, the updating metric was accuracy, the batch size was 26, and the number of total epochs was set to be 750.

To train this CNN classifier from scratch, we used the labeled ROIs of abnormal and normal mammographic images. All training data included ORG ROIs, AFF ROIs, and GAN ROIs, but validation data were only the ORG ROIs.

Table 2 Architecture of the CNN classifier.

CNN classifier	
Layer	Shape
Input: RGB image	$320 \times 320 \times 3$
Conv_3-32 + ReLU	$320 \times 320 \times 32$
MaxPooling _2	$160 \times 160 \times 32$
Conv_3-32 + ReLU	$160 \times 160 \times 32$
MaxPooling _2	$80 \times 80 \times 32$
Conv_3-64 + ReLU	$80 \times 80 \times 64$
MaxPooling _2	$40 \times 40 \times 64$
Flatten	102400
FC_64 + ReLU + Dropout (0.5)	64
FC_1	1
Output (sigmoid): [0, 1]	1

3 Experiment and Results

Our implementation of neural networks was on the Keras API backend on TensorFlow.[31] The development environment for Python was Anaconda3.

3.1 *Experiment Plan*

In this study, we applied affine transformations and GAN to augment images and compared the two augmentation methods by training a CNN classifier and assessing their classification accuracy. To the affine transformation, we first decided the padding method (Table 3).

We collected 1300 real abnormal ROIs (O_{abnorm}, "O" for original) and 1300 real normal ROIs (O_{norm}) in total. After withholding 10% for validation, there were 1170 O_{abnorm} and 1170 O_{norm}. We first augmented those data by affine transformations to obtain 1170 A_{abnorm} ("A" for affine) and 1170 A_{norm}; the details are shown in Sec. 2.2. For the three padding methods, we mark the augmented data as A^{constant}, A^{nearest}, and A^{reflect}. Then, we trained three CNN classifiers from scratch by three datasets: [1170 $A^{\text{constant}}_{\text{abnorm}}$, 1170 $A^{\text{constant}}_{\text{norm}}$], [1170 $A^{\text{nearest}}_{\text{abnorm}}$, 1170 $A^{\text{nearest}}_{\text{norm}}$] and [1170 $A^{\text{reflect}}_{\text{abnorm}}$, 1170 $A^{\text{reflect}}_{\text{norm}}$] respectively. Figure 5 shows the validation accuracy of the three CNN classifiers. Obviously, the CNN classifier trained by nearest padding AFF ROIs has the best overall performance. Therefore, we used the nearest padding AFF ROIs for the remaining experiments.

Table 3 Notations for data.

Set name	Notation for element	Meaning
ORG ROIs	$O_{\text{abnorm}}/O_{\text{norm}}$	Real abnormal/normal ROI
AFF ROIs	$A^{\text{padding}}_{\text{class}}$	Affine transformed ROI from one class (abnorm = abnormal/ norm = normal) by padding method constant/nearest/reflect
GAN ROIs	$G_{\text{abnorm}}/G_{\text{norm}}$	Synthetic abnormal/normal ROI by GAN

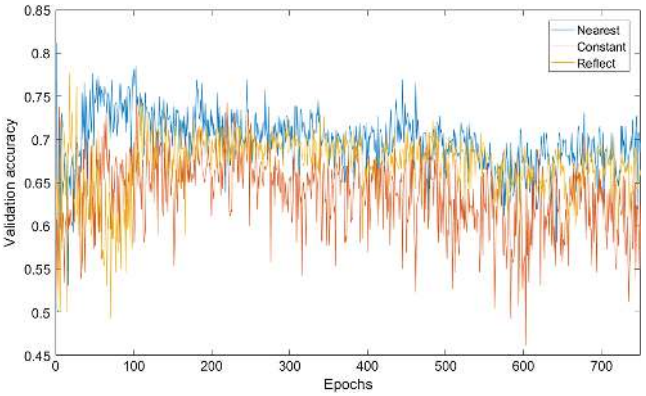

Fig. 5 Validation accuracy of CNN classifiers trained by three types of AFF ROIs.

We then used the ORG ROIs to train two generators: $\text{GAN}_{\text{abnorm}}$ and GAN_{norm} for generating GAN ROIs. As shown in Fig. 6 (GAN box), during the training process, the generator G provided synthetic ROIs to the discriminator D. D was trained to distinguish the real from the synthetic ROIs by using real and synthetic ROIs. And, once synthetic ROIs were distinguished, D gave a feedback loss to G for G's updating. Then G will generate synthetic ROIs more like the real ones. By inputting noise vectors to $\text{GAN}_{\text{abnorm}}$ and GAN_{norm}, we obtained 336 G_{abnorm} and 336 G_{norm}.

We repeated training the CNN classifier from scratch using several datasets of labeled ROIs shown in Table 4. In each set, the number of abnormal and normal ROIs was equal (Fig. 6). We used 84 O_{abnorm} and 84 O_{norm} that were had not been used in the training process as validation data to evaluate those CNN classifiers.

3.2 *Classification Results*

For training the GAN, we used 336 real abnormal ROIs to obtain the generator $\text{GAN}_{\text{abnorm}}$ and used 336 real normal ROIs to obtain the generator GAN_{norm}. Figure 7 shows some synthetic abnormal ROIs (G_{abnorm}) generated from $\text{GAN}_{\text{abnorm}}$. Then, we generated 336 G_{abnorm} and 336 G_{norm} by generators.

The results of training accuracy and validation accuracy after each training epoch (defined in Sec. 2.3, training methods, step 5; the total epochs were 750) are shown in Fig. 8. The figures make clear that sets 1, 4, and 5 performed well and set 3 was the worst. To analyze those results quantitatively, we show the stable standard deviation (SStd, which is the standard deviation of validation accuracy after 600 epochs), maximum validation accuracy (best), average validation accuracy after 600 epochs (stable), and time cost (in seconds) for each training epoch. The maximum validation accuracy can indicate the best performance of the classifier, but it may be reached fortuitously. The average validation accuracy after 600 epochs can show the stable performance of the classifier. For a good classifier, this value will be monotone increasing and converged. And SStd shows how validation accuracy varies from its average after 600 epochs. Table 5 shows these quantitative results.

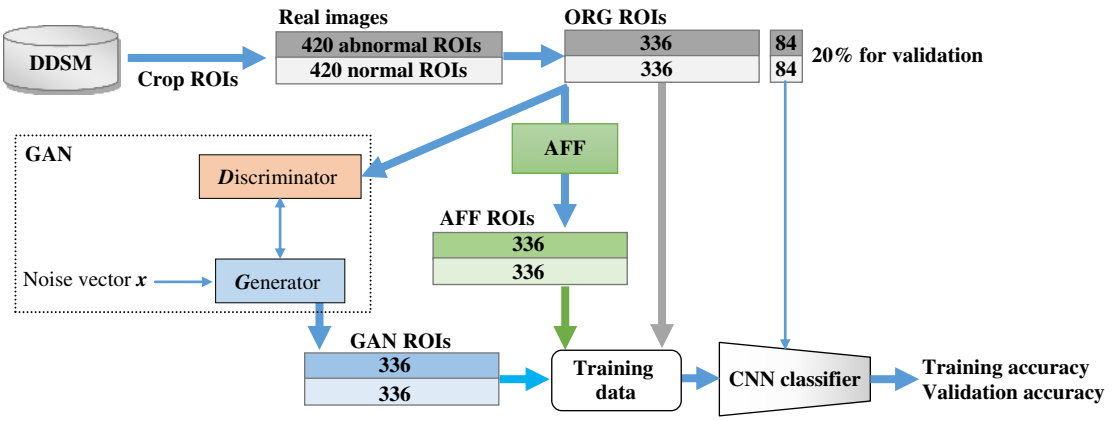

Fig. 6 Flowchart of our experiment plan. CNN classifiers were trained by data including ORG, AFF, and GAN ROIs. Validation data for the classifier were ORG ROIs that had not been used for training. The AFF box means to apply affine transformations.

Table 4 Training plans.

Classifier model	Set#	Dataset for training	Validation
CNN classifier in Table 2	1	336 O_{abnorm} labeled '1'	84 O_{abnorm} labeled '1'
		336 O_{norm} labeled '0'	84 O_{norm} labeled '0'
	2	336 G_{abnorm} labeled '1'	
		336 G_{norm} labeled '0'	
	3	336 $A^{\text{nearest}}_{\text{abnorm}}$ labeled '1'	
		336 $A^{\text{nearest}}_{\text{norm}}$ labeled '0'	
	4	336 O_{abnorm}+ 336 G_{abnorm} labeled '1'	
		336 O_{norm} + 336 G_{norm} labeled '0'	
	5	336 O_{abnorm} + 336 $A^{\text{nearest}}_{\text{abnorm}}$ labeled '1'	
		336 O_{norm} + 336 $A^{\text{nearest}}_{\text{norm}}$ labeled '0'	
	6	336 G_{abnorm}+ 336 $A^{\text{nearest}}_{\text{abnorm}}$ labeled '1'	
		336 G_{norm}+ 336 $A^{\text{nearest}}_{\text{norm}}$ labeled '0'	

Since the maximum validation accuracy may be fortuitous, the stable performance is a more reliable evaluation of a classifier. Table 5 demonstrates that:

- ORG ROIs must be added to the training set because the stable performances of sets without ORG ROIs are lower than 70%.
- By comparing set 2 with set 3, we observe that GAN-generated images could have features closer to real images than affine-transformed images. And, by comparing set 4 with set 5, we see that GAN ROIs are better than AFF ROIs for image augmentation. Inspection of the synthetic ROIs in Fig. 7 reveals some artificial components.
- Since the performance of GAN is better than affine transformation for image augmentation, GAN could be an alternative augmentation method for training CNN classifiers.

For training using only real ROIs, the validation accuracy is lower than training by adding GAN ROIs. Adding AFF ROIs can also improve the validation accuracy. Therefore, image augmentation is necessary to train CNN classifiers and since GAN performs better than affine transformation, GAN could be a good alternative option. But GAN ROIs may have features that are different from ORG ROIs because overfitting occurred. Adding ORG ROIs to the training set can help correct this problem. The images augmented by GAN or affine transformation cannot substitute for real images to train CNN classifiers because the absence of real images in the training set will cause overfitting.

4 Discussion

The hypothesis of GANs is that, in d-dimensional space, there exists a mapping function $p_{\text{data}}(x)$ from vector x to real data y; a GAN can learn and simulate the mapping function $G(x)$ by using samples from the distribution of real data. $G(x)$ is also called a generator. The ideal outcome is $G(x) = p_{\text{data}}(x)$. The maximum validation accuracy for training using GAN ROIs is about 79.8%, which shows that the generator acquired some important features from the ORG ROIs. The GAN ROIs may also have different features from those of the ORG ROIs, and so the stable accuracy is about 9% lower. Adding ORG ROIs in the training set can help correct this problem.

4.1 *Augmented-Images Analysis*

Since abnormal ROIs may contain more features than normal ROIs, we take a statistical view for comparing the real abnormal ROIs and the augmented ROIs: O_{abnorm}, $A^{\text{nearest}}_{\text{abnorm}}$, and G_{abnorm}. For each category, we use 336 samples, compute their mean, standard deviation (Std), skewness, and entropy. Then we plot the normalized values of those statistics in histograms to see their distributions. In the interest of space, we display only their Std and mean in Fig. 9.

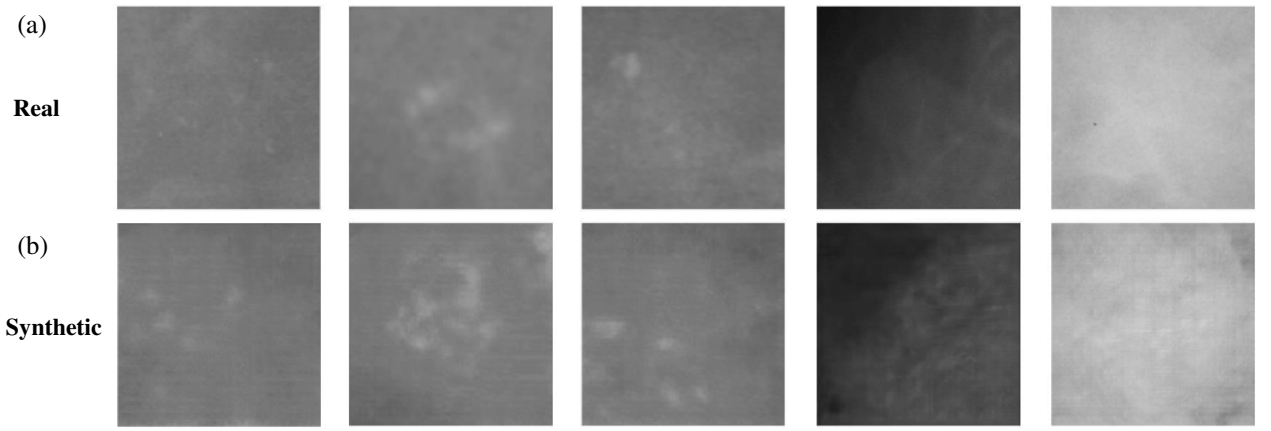

Fig. 7 (a) Real abnormal ROIs; (b) synthetic abnormal ROIs generated from GAN.

From the view of mean's distribution, GAN is more like ORG than AFF. But the view of Std's distribution shows the opposite. To quantitatively analyze difference between distributions, we calculate the Wasserstein distance[32] between two histograms. The value of the Wasserstein distance is smaller if the difference between two distributions is smaller. Wasserstein distance is equal to 0 when the two distributions are identical. Table 6 shows the Wasserstein distances of ORG ROIs versus GAN ROIs and ORG ROIs versus AFF ROIs for the four statistical descriptors.

GAN ROIs are closer than AFF ROIs to ORG ROIs in mean and entropy but farther in Std and skewness. Such results may

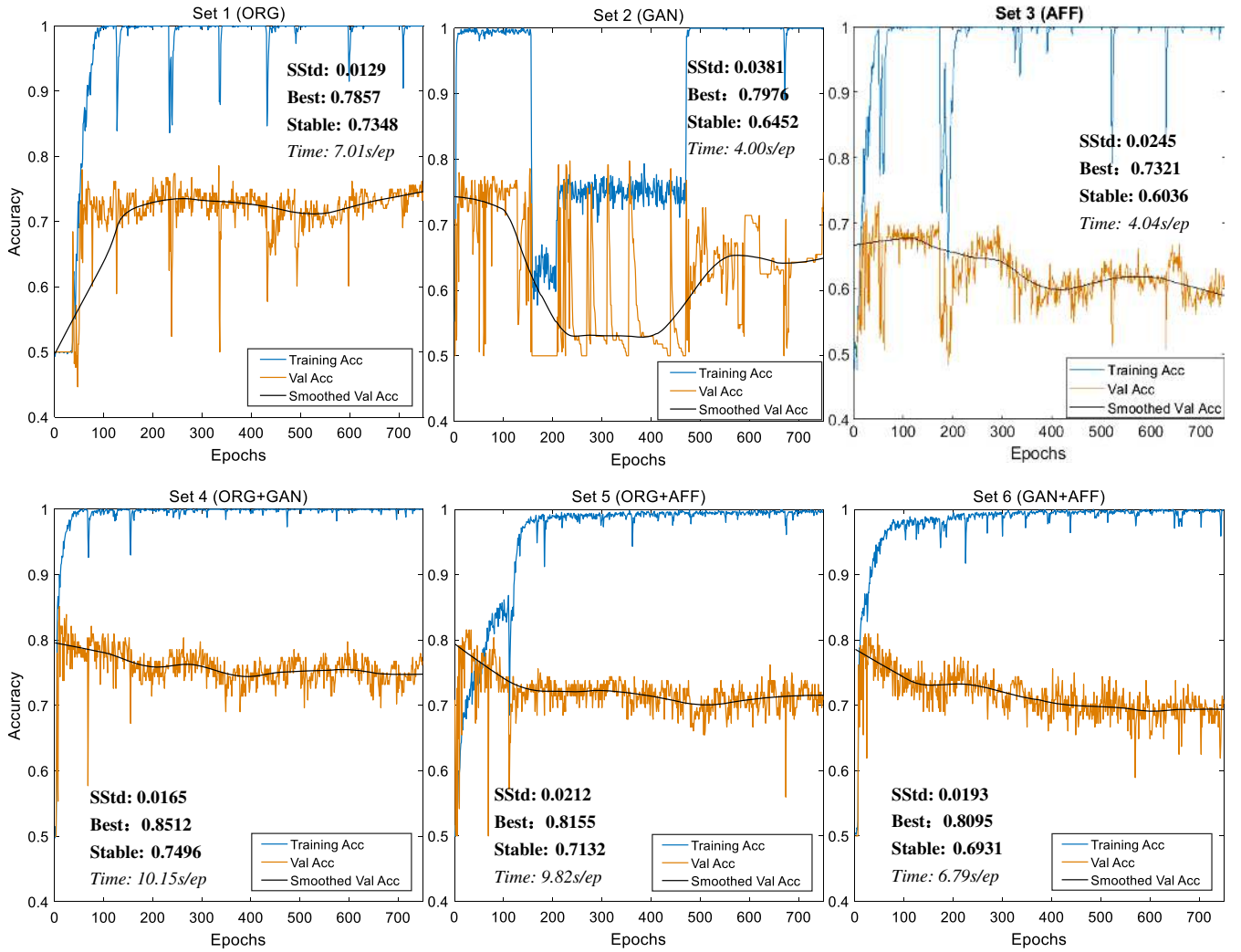

Fig. 8 Training accuracy and validation accuracy for six training datasets.

Table 5 Analysis of validation accuracy for CNN classifiers.

Set#	Best performance (%)	Stable performance (%)	SStd (%)	Time/epoch (s)
1 (ORG)	78.75	**73.48**	1.29	7.01
2 (GAN)	79.76	**64.52**	3.81	4.00
3 (AFF)	73.21	**60.36**	2.45	4.04
4 (ORG + GAN)	85.12	**74.96**	1.65	10.15
5 (ORG + AFF)	81.55	**71.32**	2.12	9.82
6 (GAN + AFF)	80.95	**69.31**	1.93	6.79

Note: The stable performance (bold values) is a more reliable index to evaluate classifiers.

Table 6 Wasserstein distance between two histograms.

Criterion	336 O_{abnorm} v.s. 336 G_{abnorm}	336 O_{abnorm} v.s. 336 $A^{nearest}_{abnorm}$
Mean	**0.083**	0.185
Std	0.100	**0.040**
Skewness	0.101	**0.047**
Entropy	**0.111**	0.456

Note: The smaller distances (bold values) mean that the two distributions are closer.

explain why GAN ROIs provide valid image augmentation. These results also suggest improvements to the GAN: we could modify the GAN to generate images having smaller Wasserstein distances to real images as measured by those statistical criteria. Actually, the most recent Wasserstein GAN[33] is designed according to a similar idea.

4.2 *Related Studies*

Since the introduction of GANs, they have been used widely in many image processing applications.[12] In medical imaging, many applications of GAN are to image segmentation.[19,21,34–37] Other applications are to medical image simulation/synthesis.[38–42]

Image synthesis is a specialty or advantage of GAN, hence, it is apt to apply GAN as an image augmentation method[43] for training classifiers and improving their detection performances. To date, however, there has been no study that uses GAN as a data-augmentation method on mammograms to train a CNN classifier for breast cancer detection. Therefore, our study fills this gap.

4.3 *Problems and Future Work*

Theoretically, a well-trained GAN could generate images having the same distributions as real images. The synthetic images will have zero Wasserstein distance to real images as measured by any statistical criteria. In that case, the performance of a CNN classifier trained by GAN ROIs will be as good as that trained by ORG ROIs. Our results, however, show that based on distribution and training performance, GAN did not meet theoretical expectations. An explanation may be found upon

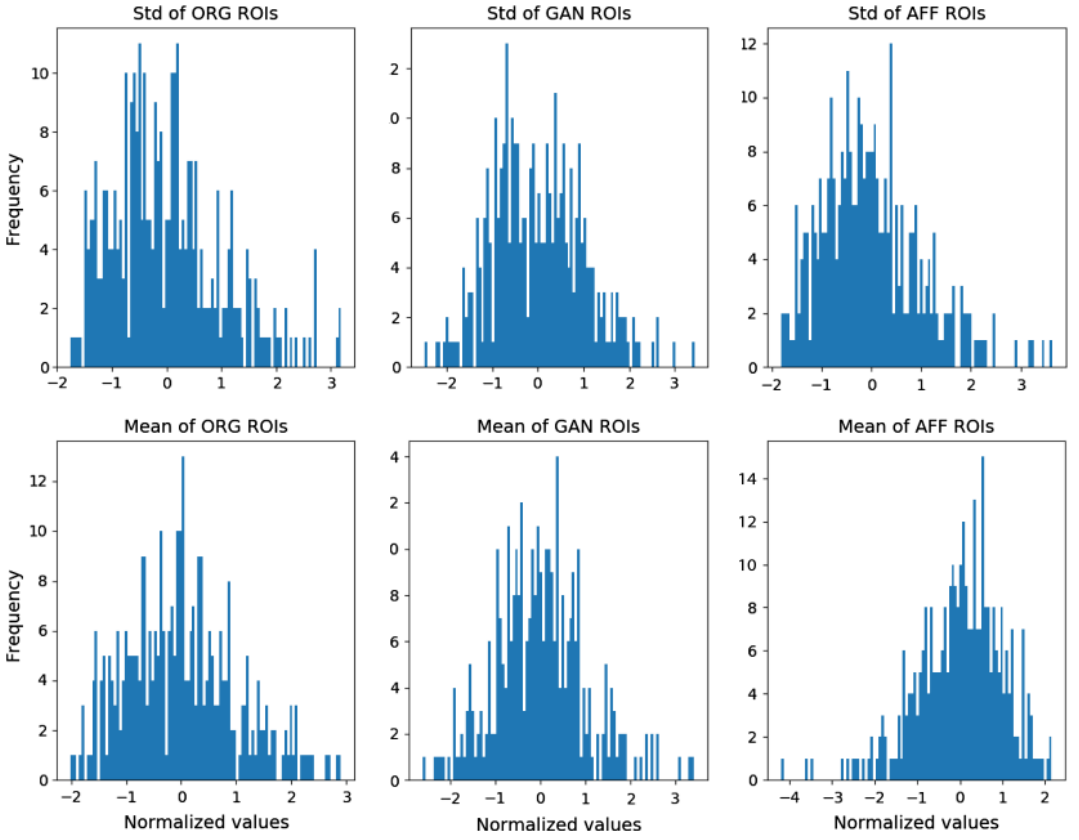

Fig. 9 Histograms of mean and skewness.

inspection of the synthetic images (Fig. 7): they have clear artifacts. One possible reason is that GAN adds some features or information not belonging to real images; that is why the distributions of the four statistical criteria of the GAN ROIs are different from those of the ORG ROIs. Those new features cause classifiers to detect abnormal features in real images and reduce the validation accuracy. A possible solution is to change the architecture of the generator or/and discriminator in GAN. In this paper, the architecture we used is DCGAN.[44] Given that ~500 architectures of GAN exist,[45] we believe that some of them can achieve a better performance for image augmentation.

In future work, we could train the classifier using transfer learning because (in addition to data augmentation) it is another important approach to deal with small training datasets. Since the DDSM provides truth labels for benign and malignant tumors, we could also perform classification for benign and malignant ROIs instead of abnormal and normal ROIs. Also, as noted above, we may examine performances of other architectures of GAN in terms of image augmentation.

5 Conclusion

In this paper, we applied GAN to generate synthetic mammograms. GAN can be used as an image augmentation method for training and to improve the performance of CNN classifiers. Our results show that, to classify the normal ROIs and abnormal (tumor) ROIs from DDSM, adding GAN-generated ROIs to the training data can help prevent overfitting (Table 5, higher stable performance). Another traditional image augmentation method—affine transformation—has poorer performance than GAN; therefore, GAN could be a preferred augmentation option. By comparing GAN ROIs with affine-transformed ROIs in their distributions of mean, standard deviation, skewness, and entropy, we found that GAN ROIs are more similar to real ROIs than affine transformed ROIs in terms of mean and entropy. Our results also show that images augmented by GAN or affine transformation cannot substitute for real images to train CNN classifiers because the absence of real images in the training set will cause overfitting with more training (stable performances lower than 70%); in other words, augmentation must mean just that.

Disclosures

The authors have no financial interests with respect to this research or publication.

References

1. R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2016," *CA Cancer J. Clin.* **66**(1), 7–30 (2016).
2. C. E. DeSantis et al., "Breast cancer statistics, 2015: convergence of incidence rates between black and white women," *CA. Cancer J. Clin.* **66**(1), 31–42 (2016).
3. V. M. Rao et al., "How widely is computer-aided detection used in screening and diagnostic mammography?" *J. Am. Coll. Radiol.* **7**(10), 802–805 (2010).
4. D. Yi et al., "Optimizing and visualizing deep learning for benign/malignant classification in breast tumors," arXiv170506362 (2017).
5. S.-C. B. Lo et al., "Artificial convolution neural network for medical image pattern recognition," *Neural Network* **8**(7–8), 1201–1214 (1995).
6. A. R. Jamieson, K. Drukker, and M. L. Giger, "Breast image feature learning with adaptive deconvolutional networks," *Proc. SPIE* **8315**, 831506 (2012).
7. D. Erhan et al., "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *Proc. Twelfth Int. Conf. Artif. Intell. and Stat.* (2009).
8. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. 25*, F. Pereira et al., Eds. Curran Associates, Inc., pp. 1097–1105 (2012).
9. H. C. Shin et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imaging* **35**(5), 1285–1298 (2016).
10. N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard?" *PLOS Comput. Biol.* **4**(1), e27 (2008).
11. I. Goodfellow et al., "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst. 27*, Z. Ghahramani et al., Eds., Curran Associates, Inc., pp. 2672–2680 (2014).
12. Y. Hong et al., "How generative adversarial networks and its variants work: an overview of GAN," arXiv:1711.05914v9 (2017).
13. C. Wang et al., "Perceptual adversarial networks for image-to-image transformation," arXiv170609138 (2017).
14. Z. Yi et al., "DualGAN: unsupervised dual learning for image-to-image translation," arXiv170402510 (2017).
15. J. Li et al., "Perceptual generative adversarial networks for small object detection," arXiv170605274 (2017).
16. C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," arXiv160904802 (2016).
17. H. Wu et al., "GP-GAN: towards realistic high-resolution image blending," arXiv170307195 (2017).
18. M. Mardani et al., "Deep generative adversarial networks for compressed sensing automates MRI," arXiv170600051 (2017).
19. Y. Xue et al., "SegAN: adversarial network with multi-scale L_1 loss for medical image segmentation," arXiv170601805 (2017).
20. D. Yang et al., "Automatic vertebra labeling in large-scale 3D CT using deep image-to-image network with message passing and sparsity regularization," arXiv170505998 (2017).
21. W. Dai et al., "SCAN: structure correcting adversarial network for organ segmentation in chest x-rays," arXiv170308770 (2017).
22. S. Guan and M. Loew, "Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks," *Proc. SPIE* **10718**, 107180X (2018).
23. M. Heath et al., "The digital database for screening mammography," in *Proc. 5th Int. Workshop on Digital Mammography*, pp. 212–218 (2000).
24. S. M. Friedewald et al., "Breast cancer screening using tomosynthesis in combination with digital mammography," *JAMA* **311**(24), 2499–2507 (2014).
25. M. Heath et al., "The digital database for screening mammography (DDSM)," 2001, http://www.eng.usf.edu/cvprg/Mammography/Database.html (February 2019).
26. A. Sharma, DDSM Utility, GitHub (2015).
27. V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, pp. 807–814 (2010).
28. N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014).
29. T. Dozat, "Incorporating Nesterov momentum into Adam" (2016).
30. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv14126980 (2014).
31. M. Abadi et al., "TensorFlow: large-scale machine learning on heterogeneous distributed systems," arXiv160304467 (2016).
32. L. Rüschendorf, "The Wasserstein distance and approximation theorems," *Probab. Theory Relat. Fields* **70**(1), 117–129 (1985).
33. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv170107875 (2017).
34. W. Zhu et al., "Adversarial deep structured nets for mass segmentation from mammograms," arXiv171009288 (2017).
35. M. Rezaei et al., "Conditional adversarial network for semantic segmentation of brain tumor," arXiv170805227 (2017).
36. J. Son, S. J. Park, and K.-H. Jung, "Retinal vessel segmentation in fundoscopic images with generative adversarial networks," arXiv170609318 (2017).
37. S. Kohl et al., "Adversarial networks for the detection of aggressive prostate cancer," arXiv170208014 (2017).
38. Y. Hu et al., "Freehand ultrasound image simulation with spatially-conditioned generative adversarial networks," *Lect. Notes Comput. Sci.* **10555**, 105–115 (2017).

39. M. J. M. Chuquicusma et al., "How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis," arXiv171009762 (2017).
40. D. Nie et al., "Medical image synthesis with context-aware generative adversarial networks," *Lect. Notes Comput. Sci.* **10435**, 417–425 (2017).
41. L. Bi et al., "Synthesis of positron emission tomography (PET) images via multi-channel generative adversarial networks (GANs)," *Lect. Notes Comput. Sci.* **10555**, 43–51 (2015).
42. J. T. Guibas, T. S. Virdi, and P. S. Li, "Synthetic medical images from dual generative adversarial networks," arXiv170901872 (2017).
43. A. J. Ratner et al., "Learning to compose domain-specific transformations for data augmentation," in *Adv. Neural Inf. Process. Syst. 30*, I. Guyon et al., Eds. Curran Associates, Inc., pp. 3239–3249 (2017).
44. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv151106434 (2015).
45. A. Hindupur, "The-GAN-zoo: a list of all named GANs!" 19 April 2017, https://deephunt.in/the-gan-zoo-79597dc8c347, (accessed February 2019).

Shuyue Guan is PhD candidate in biomedical engineering at George Washington University. His primary research interests are the applications of machine learning technologies to solve problems concerning image analysis. His current studies are the ablated tissues (lesion) detection via hyperspectral imaging and deep-learning based medical imaging. He has published 12 papers in the field of image processing and medical image analysis and presented his work in 10 exhibitions during his doctoral program.

Murray Loew is professor in the Department of Biomedical Engineering at George Washington University, Washington, DC, and director of the Medical Imaging and Image Analysis Laboratory. His interests include the development and application of image classification, thermal and hyperspectral imaging, and image fusion techniques, using machine learning for disease detection and outcome prediction. He is a fellow of SPIE, IEEE, and AIMBE, and was the inaugural Fulbright U.S.-Australia Distinguished Chair, Advanced Science and Technology (2014).