Using Generative Adversarial Networks and Transfer Learning for Breast Cancer Detection by Convolutional Neural Networks

Shuyue Guan[a], Murray Loew[a]
[a]Medical Imaging and Image Analysis Laboratory, Department of Biomedical Engineering, George Washington University, 800 22nd Street NW, Washington DC, 20052, USA

ABSTRACT

In the U.S., breast cancer is diagnosed in about 12% of women during their lifetime and it is the second leading reason for women's death. Since early diagnosis could improve treatment outcomes and longer survival times for breast cancer patients, it is significant to develop breast cancer detection techniques. The Convolutional Neural Network (CNN) can extract features from images automatically and then perform classification. To train the CNN from scratch, however, requires a large number of labeled images, which is infeasible for some kinds of medical image data such as mammographic tumor images. In this paper, we proposed two solutions to the lack of training images. 1)To generate synthetic mammographic images for training by the Generative Adversarial Network (GAN). Adding GAN generated images made to train CNN from scratch successful and adding more GAN images improved CNN's validation accuracy to at most (best) 98.85%. 2)To apply transfer learning in CNN. We used the pre-trained VGG-16 model to extract features from input mammograms and used these features to train a Neural Network (NN)-classifier. The stable average validation accuracy converged at about 91.48% for classifying abnormal vs. normal cases in the DDSM database. Then, we combined the two deep-learning based technologies together. That is to apply GAN for image augmentation and transfer learning in CNN for breast cancer detection. To the training set including real and GAN augmented images, although transfer learning model did not perform better than the CNN, the speed of training transfer learning model was about 10 times faster than CNN training. Adding GAN images can help training avoid over-fitting and image augmentation by GAN is necessary to train CNN classifiers from scratch. On the other hand, transfer learning is necessary to be applied for training on pure real images. To apply GAN to augment training images for training CNN classifier obtained the best classification performance.

Keywords: breast mass classification, deep learning, convolutional neural networks, generative adversarial networks, transfer learning, mammogram, image augmentation, computer-aided diagnosis

1. INTRODUCTION

Breast cancer is the second leading cause of death among U.S 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 for the patients [3]. These traditional CAD tools, however, have a variety of drawbacks because they rely on manually designed features. For example, hand-crafted features tend to be domain-specific, and the process of feature design can be tedious, difficult, and non-generalizable [4]. In recent years, developments in machine learning have provided alternative methods 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, however, to obtain a sufficient number of images to train a CNN classifier is difficult because the true positives are scarce in the datasets and expert labeling is expensive [9].

The shortcomings of an insufficient number of images to train a classifier are well-known [8,10], so it is worthwhile to apply **image augmentation** to create new training images and thus to improve the performance of a CNN classifier. Like CNN, the Generative Adversarial Network (GAN) is a state-of-the-art neural network-based learning technique in the field of deep learning [11] introduced by Goodfellow *et al.* in 2014 [12]. Many novel applications in the field of image processing has been provided via GAN, for example, image translation [13,14], object detection [15], super-resolution [16] and image blending [17]. Also for the medical imaging, various GAN are also developed recently such as GANCS [18] for MRI reconstruction, SegAN [19], DI2IN [20] and SCAN [21] for medical image segmentation. An image augmentation method is to generate synthetic images by 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. Therefore, GAN is a good candidate as such image augmentation method for augmenting the training dataset. We name the original images **ORG images** and the augmented images generated from GAN **GAN images** in the rest of this paper.

Another solution to deal with the lack of training images is to reuse a pre-trained CNN model that has been trained with very large image datasets from other fields as the feature extractor and re-train (fine-tune) such a model using a limited number of labeled medical images [22]. This approach is also called **transfer learning**, which has been successfully applied to various computer vision questions [23–25]. In fact, some results of transfer learning are counterintuitive: previous studies for the pulmonary embolism and melanocytic lesion detection [22,26] show that the features (connection weights in the CNN) learned from natural images could be transferred to medical images, even if the target images greatly differ from the pre-trained source images.

Previous studies have applied various machine learning methods for breast cancer/tumor detection using mammograms [27]. The Digital Database for Screening Mammography (DDSM) [28] are the most commonly used public mammogram databases. Some studies used the traditional automatic feature extraction (not manual extraction) techniques, such as Gabor filter, fractional Fourier transform and Gray Level Co-Occurrence Matrix (GLCM), to obtain features and then applied SVM or other classifier to do classification [29–33]. Neural networks were also used as classifiers [34,35]. And some studies applied CNN to generate features from mammographic images [36–39]. Some of these studies used pre-trained CNN as applications of transfer learning. In our study, we have tested both GAN for image augmentation and transfer learning to improve the performance of CNN classifier to breast cancer detection in mammograms. Specifically, we tested three training strategies on DDSM: 1) trained a CNN from scratch; 2) applied the pre-trained VGG-16 model [40] to extract features from input images and used these features to train a Neural Network (NN)-classifier; 3) added GAN images in training set and repeated experiments in (1) and (2).

2. MATERIALS AND METHODS

2.1 The Mammogram Databases and Image Pre-processing

Mammography is the process of using low-energy X-rays to examine the human breast for diagnosis and screening. There are two main angles to get the X-ray images: the cranio-caudal (CC) view and the mediolateral-oblique (MLO) view. The goal of mammography is the early detection of breast cancer [41], typically through detection of masses or abnormal regions from the formed X-ray images. Usually, such abnormal regions are spotted by doctors or expert radiologists. In this study, we used mammogram from the Digital Database for Screening Mammography (DDSM) [28]. The DDSM is a widely used mammographic images resource by the U.S. Mammographic Image Analysis Research Community. 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 approximately 2,620 cases in total: 695 normal cases, 1925 abnormal cases (914 malignant\cancers cases, 870 benign cases 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.

We downloaded all mammographic images from DDSM's official website (http://marathon.csee.usf.edu/Mammography /Database.html). Since images in DDSM are compressed in LJPEG format, to decompress and convert these images, we used the DDSM Utility [42]. 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 implemented all pre-processing tasks in MATLAB. We used the regions of interest of images (ROIs) instead of entire images to train our neural-network models. 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.
- For **normal ROIs**, they are also rectangle-shape images and their sizes are approximately the average size of abnormal ROIs. In DDSM, the average size of abnormal ROIs is 506.02×503.90 pixels, so the cropping size for normal ROIs was chosen to be 505×505 pixels. Their locations are selected randomly on normal breast areas. In this study, we cropped only one ROI from an entire normal breast image.

The sizes of abnormal ROIs vary with abnormality boundaries. Since the neural-network models require 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 made them to RGB (3-layer cubes) by duplication (Fig. 1). These images cropped from mammogram are **ORG ROIs**.

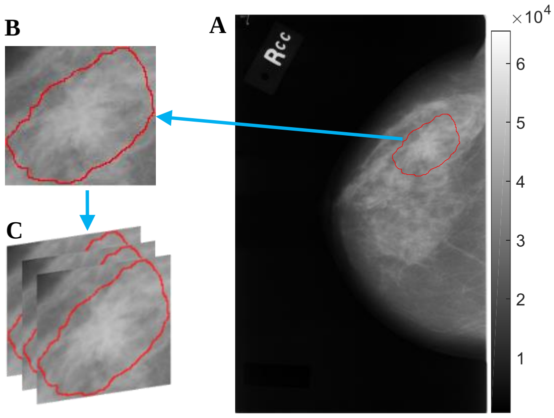

Fig. 1. (A) A mammographic image from DDSM rendered in grayscale; (B) Cropped ROI by the given truth abnormality boundary; (C) Convert Grey to RGB image by duplication.

2.2 GAN Image Augmentation

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. 2). Formally, in d-dimension space, for $x \in R^d$, $y = p_{data}(x)$ 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; and sample z comes from G, it is a synthetic one. Another neural network **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 [12], with value function $V(G,D)$:

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

This min-max problem has a global optimum (Nash equilibrium) solution for $G(x) = p_{data}(x)$. That is the goal to find the distribution of real data. At equilibrium, 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 we used was a noise vector having 100 elements from a Gaussian distribution : $N(0,1)$. The key point of a well-trained GAN is that it could generate seemingly real-like data samples by giving noise vectors. To train a GAN, we used 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 show in

Table 1. The generator consists of four up-sampling layers to double the size of image and five convolutional layers. The activation function for each layer is the ReLU function [43] except the last one for output, which is tanh function. The function of generator is to transform a 100-length vector to a 320x320x3 image. The input of discriminator is a 320x320x3 image and its output is a value between 0 and 1, which '0' stands for the synthetic image and '1' for the real one. Like a typical CNN, the discriminator has four convolutional layers with max-pooling layers and one FC layer. The activation function for each convolutional layer is also the ReLU function and the last one for output is sigmoid function, which maps the output value to the range of [0, 1].

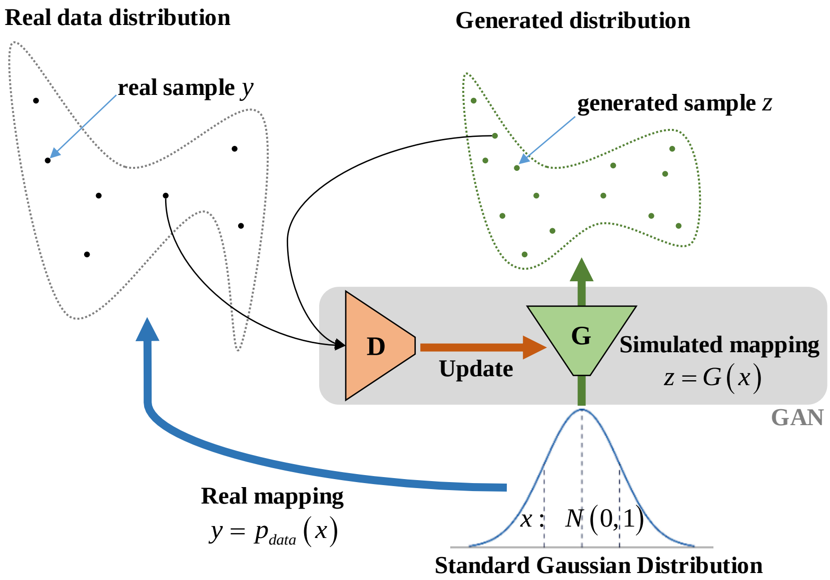

Fig. 2. The principle of GAN.

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 size of filters is 2×2-pixel window, stride 2. And FC_n means a fully-connected layer having *n* units. Dropout layer [44] randomly set a fraction rate of input units to 0 for the next layer at every updating during training; it could help the networks avoid overfitting. Our training optimizer is Nadam [45] using default parameters (except the learning rate changed to 1e-4), the loss function is Binary Cross Entropy, the updating metric is Accuracy, the batch size is 30 and the number of total epochs is 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 100-length noise vectors to generator to obtain synthetic images.
- Step 3: To 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 100-length noise vectors to generator to obtain synthetic images and label them as '1'. Then, input these synthetic images to 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 generator are changed; weights in discriminator are fixed.
- Step 5: Repeat Step 2 to Step 4 until all real images have been used once, that counts one **epoch**. When the number of epochs reaches a certain value, training stops.

Actually, for the Step 5, the ideal situation to stop training is when the classification accuracy of discriminator converges to 50%. It means the discriminator no longer can distinguish the real images and synthetic images generated from a well-trained generator. The discriminator plays a role as an assistant in GAN. After training, we will use the generator neural networks to generate synthetic images for usage next.

Table 1. The architecture of generator and discriminator neural networks.

Generator	
Layer	Shape
input: 100-length vector	100
FC_(256x20x20) + ReLU	102400
Reshape to 20x20x256	20x20x256
Normalization + Up-sampling	40x40x256
Conv_3-256 + ReLU	40x40x256
Normalization + Up-sampling	80x80x256
Conv_3-128 + ReLU	80x80x128
Normalization + Up-sampling	160x160x128
Conv_3-64 + ReLU	160x160x64
Normalization + Up-sampling	320x320x64
Conv_3-32+ ReLU	320x320x32
Normalization + Conv_3-3+ ReLU	320x320x3
output (tanh): [-1, 1]	320x320x3

Discriminator	
Layer	Shape
input: RGB image	320x320x3
Conv_3-32 + ReLU	320x320x32
MaxPooling_2 + Dropout (0.25)	160x160x32
Conv_3-64 + ReLU	160x160x64
MaxPooling_2 + Dropout (0.25)	80x80x64
Conv_3-128 + ReLU	80x80x128
MaxPooling_2 + Dropout (0.25)	40x40x128
Conv_3-256 + ReLU	40x40x256
MaxPooling_2 + Dropout (0.25)	20x20x256
Flatten	102400
FC_1	1
output (sigmoid): [0, 1]	1

2.3 To Train the CNN from Scratch

Actually, a CNN was designed as the discriminator in GAN and its function is to distinguish real and synthetic mammographic ROIs. We also built a CNN to classify abnormal ROIs and normal ROIs. As shown in Table 2, this CNN classifier consists of three convolutional layers with max-pooling layers and two FC layers. The activation function for each layer is the ReLU function except the last one for output. The output layer uses a sigmoid function, which maps the output value to the range of [0, 1]. Its input is the image in size 320×320-pixel. Since the sigmoid function was used in the output layer, the predicted outcome from the CNN classifier is a value between 0 and 1. By default, the classification threshold is 0.5, meaning that if the value is less than 0.5 it will be considered as "0" (normal), otherwise it will be considered as "1" (abnormal). The optimizer for training is Nadam using default parameters [45] (except the learning rate changed to 1e-4), the loss function is Binary Cross Entropy, the updating metric is Accuracy, the batch size is 26 and the number of total epochs is set to be 750. To train this CNN classifier from scratch, we used the labeled ROIs of abnormal and normal mammographic images.

Table 2. The architecture of CNN classifier.

CNN classifier	
Layer	Shape
input: RGB image	320x320x3
Conv_3-32 + ReLU	320x320x32
MaxPooling _2	160x160x32
Conv_3-32 + ReLU	160x160x32
MaxPooling _2	80x80x32
Conv_3-64 + ReLU	80x80x64
MaxPooling _2	40x40x64
Flatten	102400
FC_64 + ReLU + Dropout (0.5)	64
FC_1	1
output (sigmoid): [0, 1]	1

2.4 Transfer Learning: Features Extraction by Pre-trained VGG-16 network

The structure of CNN in transfer learning was combined the 13 convolutional layers in pre-trained VGG-16 model [40] with a simple FC layer (Table 3).

Table 3. CNN architecture for transfer learning

CNN classifier with Transfer Learning		
Layer		
		input: RGB image
VGG-16	Conv block 1	Conv_3-64 + ReLU
		Conv_3-64 + ReLU
		MaxPool_2
	Conv block 2	Conv_3-128 + ReLU
		Conv_3-128 + ReLU
		MaxPool_2
	Conv block 3	Conv_3-256 + ReLU
		Conv_3-256 + ReLU
		Conv_3-256 + ReLU
		MaxPool_2
	Conv block 4	Conv_3-512 + ReLU
		Conv_3-512 + ReLU
		Conv_3-512 + ReLU
		MaxPool_2
	Conv block 5	Conv_3-512 + ReLU
		Conv_3-512 + ReLU
		Conv_3-512 + ReLU
		MaxPool_2
FC_256 + ReLU (with Dropout = 0.5)		
output (sigmoid): [0, 1]		

As shown in Table 3, all the weights in five convolutional blocks (the blue background layers) were imported from the pre-trained VGG-16 model and not changed (or called weights frozen) during the training of this CNN. Only weights in the FC layer were randomly initialized and updated by training. Thus, such training process can be seen as that the VGG-16 extracts features from input image and then these features were used to train a FC NN-classifier.

3. EXPERIMENT AND RESULTS

Our implementation of neural networks was on the Keras API backend on TensorFlow [46]. The development environment for Python was Anaconda3.

3.1 Experiment Plan

Table 4. Notations for data.

Set name	Notation for element	Meaning
ORG ROIs	O_{abnorm} / O_{norm}	Real abnormal/normal ROI
GAN ROIs	G_{abnorm} / G_{norm}	Synthetic abnormal/normal ROI by GAN

In this study, we collected 1300 original (real) abnormal ROIs (O_{abnorm} , 'O' for original) and 1300 original normal ROIs (O_{norm}) in total. After taking off 10% for validation, there are 1170 O_{abnorm} and 1170 O_{norm} . We firstly did the data augmentation to 1170 O_{abnorm} and 1170 O_{norm} by GAN. We used the 1170 O_{abnorm} and 1170 O_{norm} to train two generators respectively: GAN_{abnorm} and GAN_{norm} for generating **GAN ROIs**. As shown in Fig. 3 (GAN box), during the training process, the generator G provided synthetic ROIs to discriminator D . D was trained to distinguish the real from the synthetic ROIs by real and synthetic ROIs. And once synthetic ROIs were distinguished, D gave feedback loss to G for G's updating. Then G will generate synthetic ROIs more like the real ones. By inputting noise vectors to GAN_{abnorm} and GAN_{norm} , we obtained G_{abnorm} and G_{norm} . Fig. 4 shows some synthetic abnormal ROIs (G_{abnorm}) generated from GAN_{abnorm} .

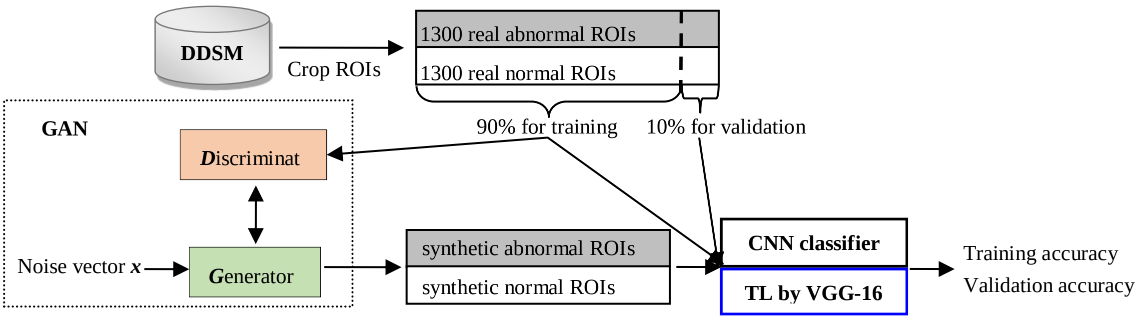

Fig. 3. The flowchart of our experiment plan. CNN classifiers were trained by data including ORG and GAN ROIs. Validation data for the classifier were ORG ROIs that were never used for training. The ORG and GAN ROIs were also used to Transfer Learning by pre-trained VGG-16 model.

Table 5. Training plans.

Set#	Dataset for training	Validation	Classifier Model	
1	1170 O_{abnorm} labeled '1' 1170 O_{norm} labeled '0'	130 O_{abnorm} labeled '1' 130 O_{norm} labeled '0'	CNN in Table 2	TL model in Table 3
2	1170 G_{abnorm} labeled '1' 1170 G_{norm} labeled '0'			
3	1170 O_{abnorm} + 1170 G_{abnorm} labeled '1' 1170 O_{norm} + 1170 G_{norm} labeled '0'			
4	1170 O_{abnorm} + 2340 G_{abnorm} labeled '1' 1170 O_{norm} + 2340 G_{norm} labeled '0'			

We repeated training the **CNN classifier** and the **transfer learning (TL) model** from scratch using different datasets of labeled ROIs shown in Table 5. During the training, there was no any data augmentation applied. In each set, the number of abnormal ROIs and normal ROIs is equal. We used 130 O_{abnorm} and 130 O_{norm} that were never used in the training process as validation data to evaluate those CNN classifiers. We generated 1170 G_{abnorm} and 1170 G_{norm} from GAN for training Set 2 and combined with 2340 ORG ROIs for Set 3. In the Set 3, the number of ORG ROIs and GAN ROIs are equal. In addition, we generated double number of GAN ROIs as ORG ROIs and put them together in Set 4.

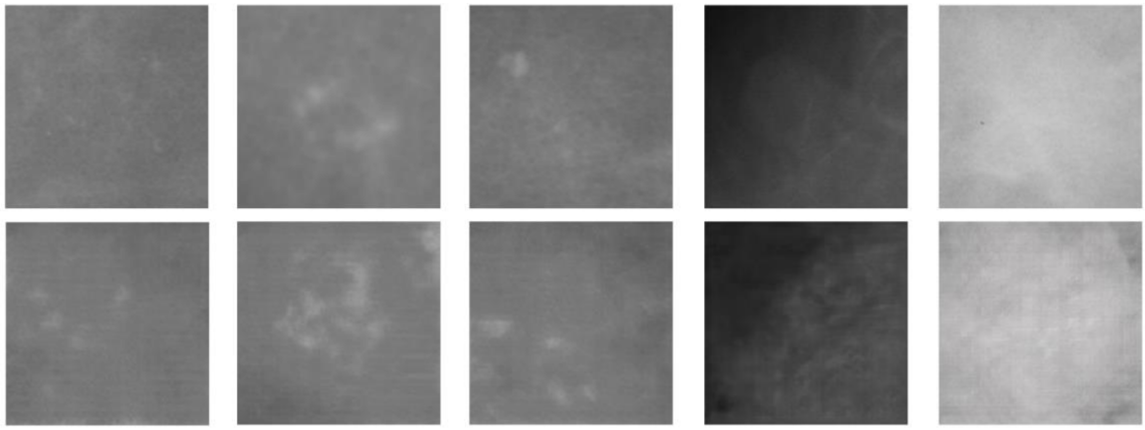

Fig. 4. (Top row) Real **abnormal** ROIs; (Bottom row) synthetic **abnormal** ROIs generated from GAN.

3.2 Classification Results

Specifically, to train GAN, we used 1170 O_{abnorm} to obtain the generator GAN_{abnorm}, and used 1170 O_{norm} to obtain the generator GAN_{norm}. Fig. 4 shows some synthetic abnormal ROIs (G_{abnorm}) generated from GAN_{abnorm}. Then, we generated G_{abnorm} and G_{norm} by generators.

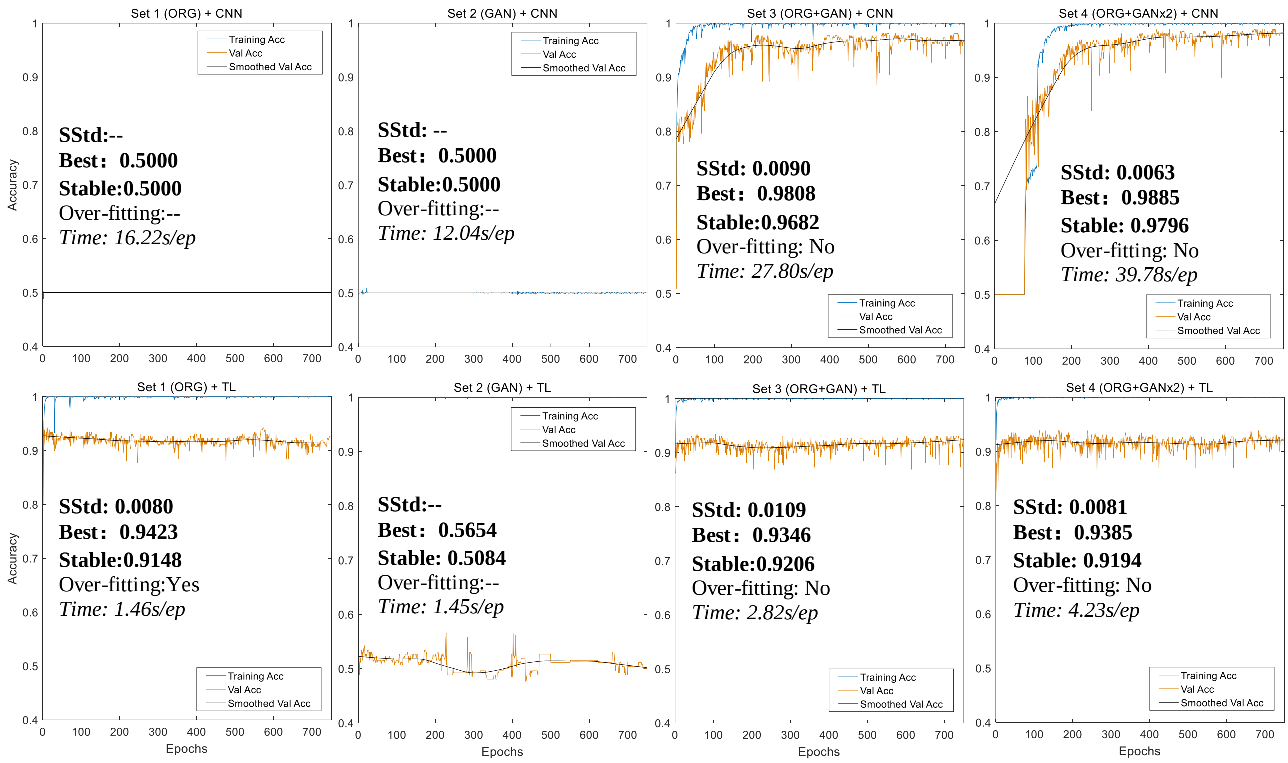

Fig. 5. Training accuracy and validation accuracy for four training datasets.

The results of training accuracy and validation accuracy after each training epoch (it is defined in 2.2, training methods, Step 5; the total epochs are 750) are shown in Fig. 5. By looking the figures, Set 3 and 4 perform well and Set 1 and 2 are worse except Set 1 using transfer learning. 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 whether the over-fitting occurred. 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. The criterion for **occurrence of over-fitting** is defined by the value: average validation accuracy after 400 epochs minus (-) average validation accuracy before 400 epochs; if it is negative, then we consider that the over-fitting occurred because of the decreasing of validation accuracy during training.

Since the maximum validation accuracy may be fortuitous, the stable performance during training is more reliable to evaluate a classifier. The results in Fig. 5 demonstrate that:

- Pure ORG ROIs or GAN ROIs cannot train the CNN classifier successfully. To train CNN from scratch, adding GAN ROIs made the CNN classifier training successful. Additionally, by comparing the two results of Set 3 and Set 4, more added GAN ROIs improved CNN's performance.
- By comparing the two results of Set 1, transfer learning (TL) model successfully improved the accuracy of classification a lot. But to compare TL used to Set 1 and Set 2, pure GAN ROIs also cannot train the transfer learning model successfully.
- To transfer learning model, by examining Set 1, 3 and 4, adding GAN ROIs did not improve validation accuracies very much, however, prevented the training from over-fitting.
- By comparing the results of CNN and TL for Set 3 and 4, adding GAN ROIs have more benefit to improved CNN's performance remarkably than to the TL. TL has the advantage on speed – TL was running about 10 times faster than CNN.

Overall, to train TL by only ORG ROIs, the validation accuracy is as good as training by adding GAN ROIs. But adding GAN ROIs can help avoid over-fitting. Image augmentation by GAN is necessary to train CNN classifiers from scratch. On the other hand, TL is necessary to be applied for training on pure ORG ROIs. To apply GAN to augment training images for training CNN classifier obtained the best classification performance. Then, to decrease the time cost of training, TL could be also applied to the augmented dataset.

4. DISCUSSION

As we discussed in Section 2.2, the ideally theoretical outcome of GAN is $G(x) = p_{data}(x)$. If so, the performance of CNN classifier trained by GAN ROIs will be as good as by ORG ROIs. Our results, however, show that GAN did not correspond with theoretical expectations. Opposite to ORG ROIs, pure GAN ROIs cannot train the transfer learning model successfully by comparing TL used to Set 1 and Set 2. The problem could be found by looking the synthetic images (Fig. 4): they have clear artificial flavors. One possible reason is that GAN adds some features or information not belonging to real images. Those new features disturb classifiers to detect abnormal features in real images. The possible solution is to change the architecture of generator or/and discriminator in GAN. In this paper, the architecture we used is DCGAN [47]. Recently, there are about 500 architectures of GAN [48]. We believe that some of them can achieve a better performance to train CNN from scratch.

We reviewed several recent studies highly related to ours. These studies (Table 6) applied transfer learning in CNN to detect breast cancer/abnormality based on mammogram. By comparison with these studies, we used many more mammographic images for training and testing the CNN classifiers and a distinct pre-trained model. The main difference is about the classifier and image augmentation by GAN. Our one-FC layer NN-classifier has simpler architecture and could be integrated with pre-trained convolutional layers as one complete CNN. The stable classification accuracies of our proposed model for abnormal vs normal cases on mammograms are competitive to other studies.

Since the GAN were introduced, it has been widely used in many image processing applications [11]. In medical imaging, many applications of GAN are segmentation [19,21,49–52]. And some studies are about medical image simulation/synthesis [53–57]. Image synthesis is a specialty or advantage of GAN, hence, it is apt to apply GAN as an image augmentation method [58] for training classifiers and improving their detection performances. As far as we aware there is no study about using GAN as data augmentation method on mammogram to train CNN classifier or transfer learning model for breast cancer detection. Therefore, our study fills this gap.

Since the DDSM provides truth labels for benign and malignant tumors, in future works, we could also do classification for benign and malignant ROIs instead of abnormal and normal ROIs. We could try to recognize the abnormal areas in whole mammographic images. By using the RCNN [59], we could recognize the abnormalities on mammographic images and draw boundaries (or rectangle region proposals) on such areas automatically. These regions do not have to be very high accuracy because they just provide another kind of reference for doctors to make decisions. We could use other pre-trained models and compare to their performances. In the research field of deep learning, VGG-16 appeared early but its depth (total number of layers is 23) is relatively shallow compared to new models, such as InceptionV3 (159 layers) [60], ResNet50 (168 layers) [61] and InceptionResNetV2 (572 layers) [62]. It will be interesting to see performances of breast cancer detection by using very deep CNNs. And we may also examine performances of other architectures of GAN in terms of image augmentation.

Table 6. Comparison of related studies.

Main method	**# of images**	**Accuracy %**
Pre-trained CNN on LSVRC datasets & Fine-tuning + Two-step decision[37]	600	(Ben-Mal) 96.7
Pre-trained CNN with hand crafted features + RF[38]	410	(Ben-Mal) 91.0
Pre-trained AlexNet +Sparse MIL[36]	410	(Mal-nonMal) 90.0
Pre-trained VGG-16 + one FC layer by ORG ROIs (Ours)	2600	(Abnorm-Norm) 91.5
Pre-trained VGG-16 + one FC layer by ORG+GAN ROIs (Ours)	2600	(Abnorm-Norm) 92.1
CNN by ORG + *double* GAN ROIs (Ours)	2600	(Abnorm-Norm) 98.0

5. CONCLUSIONS

In this paper, we proposed GAN to 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 in training data can help the classifier prevent from over-fitting and the validation accuracy using mixture ROIs reached at most (best) 98.85%. Therefore, GAN could be promising image augmentation method. To transfer learning in CNN for breast cancer detection, our results show that the pre-trained CNN model (VGG-16) can automatically extract features from mammographic images, and a good NN-classifier (achieves stable average validation accuracy about 91.48% for classifying abnormal vs. normal cases in the DDSM database) can be trained by only real ROIs. In addition, we have done the study of combining the two deep-learning-based technologies together. That is to apply GAN for image augmentation and then use transfer learning in CNN for detection. Although to train the transfer learning model by adding GAN ROIs did not perform better than to train the CNN by adding GAN ROIs, the speed of training transfer learning model was about 10 times faster than CNN training. In summary, adding GAN ROIs can help training avoid over-fitting and image augmentation by GAN is necessary to train CNN classifiers from scratch. On the other hand, transfer learning is necessary to be applied for training on pure ORG ROIs. To apply GAN to augment training images for training CNN classifier obtained the best classification performance.

REFERENCES

[1] Siegel, R. L., Miller, K. D. and Jemal, A., “Cancer statistics, 2016,” CA. Cancer J. Clin. **66**(1), 7–30 (2016).

[2] DeSantis, C. E., Fedewa, S. A., Goding Sauer, A., Kramer, J. L., Smith, R. A. and Jemal, A., “Breast cancer statistics, 2015: Convergence of incidence rates between black and white women,” CA. Cancer J. Clin. **66**(1), 31–42 (2016).

[3] Rao, V. M., Levin, D. C., Parker, L., Cavanaugh, B., Frangos, A. J. and Sunshine, J. H., “How Widely Is Computer-Aided Detection Used in Screening and Diagnostic Mammography?,” J. Am. Coll. Radiol. **7**(10), 802–805 (2010).

[4] Yi, D., Sawyer, R. L., Cohn III, D., Dunnmon, J., Lam, C., Xiao, X. and Rubin, D., “Optimizing and Visualizing Deep Learning for Benign/Malignant Classification in Breast Tumors,” ArXiv170506362 Cs (2017).

[5] Lo, S.-C. B., Chan, H.-P., Lin, J.-S., Li, H., Freedman, M. T. and Mun, S. K., “Artificial convolution neural network for medical image pattern recognition,” Neural Netw. **8**(7–8), 1201–1214 (1995).
[6] Jamieson, A. R., Drukker, K. and Giger, M. L., “Breast image feature learning with adaptive deconvolutional networks,” presented at Medical Imaging 2012: Computer-Aided Diagnosis, 23 February 2012, 831506, International Society for Optics and Photonics.
[7] Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S. and Vincent, P., “The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training” (2009).
[8] Krizhevsky, A., Sutskever, I. and Hinton, G. E., “ImageNet Classification with Deep Convolutional Neural Networks,” [Advances in Neural Information Processing Systems 25], F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 1097–1105 (2012).
[9] Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R. M., “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] Pinto, N., Cox, D. D. and DiCarlo, J. J., “Why is Real-World Visual Object Recognition Hard?,” PLOS Comput. Biol. **4**(1), e27 (2008).
[11] Hong, Y., Hwang, U., Yoo, J. and Yoon, S., “How Generative Adversarial Networks and its variants Work: An Overview of GAN,” ArXiv171105914 Cs (2017).
[12] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., “Generative Adversarial Nets,” [Advances in Neural Information Processing Systems 27], Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2672–2680 (2014).
[13] Wang, C., Xu, C., Wang, C. and Tao, D., “Perceptual Adversarial Networks for Image-to-Image Transformation,” ArXiv170609138 Cs (2017).
[14] Yi, Z., Zhang, H., Tan, P. and Gong, M., “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation,” ArXiv170402510 Cs (2017).
[15] Li, J., Liang, X., Wei, Y., Xu, T., Feng, J. and Yan, S., “Perceptual Generative Adversarial Networks for Small Object Detection,” ArXiv170605274 Cs (2017).
[16] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. and Shi, W., “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” ArXiv160904802 Cs Stat (2016).
[17] Wu, H., Zheng, S., Zhang, J. and Huang, K., “GP-GAN: Towards Realistic High-Resolution Image Blending,” ArXiv170307195 Cs (2017).
[18] Mardani, M., Gong, E., Cheng, J. Y., Vasanawala, S., Zaharchuk, G., Alley, M., Thakur, N., Han, S., Dally, W., Pauly, J. M. and Xing, L., “Deep Generative Adversarial Networks for Compressed Sensing Automates MRI,” ArXiv170600051 Cs Stat (2017).
[19] Xue, Y., Xu, T., Zhang, H., Long, R. and Huang, X., “SegAN: Adversarial Network with Multi-scale L_1 Loss for Medical Image Segmentation,” ArXiv170601805 Cs (2017).
[20] Yang, D., Xiong, T., Xu, D., Huang, Q., Liu, D., Zhou, S. K., Xu, Z., Park, J., Chen, M., Tran, T. D., Chin, S. P., Metaxas, D. and Comaniciu, D., “Automatic Vertebra Labeling in Large-Scale 3D CT using Deep Image-to-Image Network with Message Passing and Sparsity Regularization,” ArXiv170505998 Cs (2017).
[21] Dai, W., Doyle, J., Liang, X., Zhang, H., Dong, N., Li, Y. and Xing, E. P., “SCAN: Structure Correcting Adversarial Network for Organ Segmentation in Chest X-rays,” ArXiv170308770 Cs (2017).
[22] Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B. and Liang, J., “Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?,” IEEE Trans. Med. Imaging **35**(5), 1299–1312 (2016).
[23] Sharif Razavian, A., Azizpour, H., Sullivan, J. and Carlsson, S., “CNN features off-the-shelf: an astounding baseline for recognition,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 806–813 (2014).
[24] Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A. and Carlsson, S., “From generic to specific deep representations for visual recognition,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 36–45 (2015).
[25] Penatti, O. A., Nogueira, K. and dos Santos, J. A., “Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 44–51 (2015).
[26] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. and Thrun, S., “Dermatologist-level classification of skin cancer with deep neural networks,” Nature **542**(7639), 115–118 (2017).

[27] Ganesan, K., Acharya, U. R., Chua, C. K., Min, L. C., Abraham, K. T. and Ng, K. H., "Computer-Aided Breast Cancer Detection Using Mammograms: A Review," IEEE Rev. Biomed. Eng. **6**, 77–98 (2013).
[28] Heath, M., Bowyer, K., Kopans, D., Moore, R. and Kegelmeyer, W. P., "The digital database for screening mammography," Proc. 5th Int. Workshop Digit. Mammogr., 212–218, Medical Physics Publishing (2000).
[29] Khan, S., Hussain, M., Aboalsamh, H. and Bebis, G., "A comparison of different Gabor feature extraction approaches for mass classification in mammography," Multimed. Tools Appl. **76**(1), 33–57 (2017).
[30] Raghavendra, U., Rajendra Acharya, U., Fujita, H., Gudigar, A., Tan, J. H. and Chokkadi, S., "Application of Gabor wavelet and Locality Sensitive Discriminant Analysis for automated identification of breast cancer using digitized mammogram images," Appl. Soft Comput. **46**, 151–161 (2016).
[31] Khan, S., Hussain, M., Aboalsamh, H., Mathkour, H., Bebis, G. and Zakariah, M., "Optimized Gabor features for mass classification in mammography," Appl. Soft Comput. **44**, 267–280 (2016).
[32] Zhang, Y.-D., Wang, S.-H., Liu, G. and Yang, J., "Computer-aided diagnosis of abnormal breasts in mammogram images by weighted-type fractional Fourier transform," Adv. Mech. Eng. **8**(2), 1687814016634243 (2016).
[33] Narváez, F., Alvarez, J., Garcia-Arteaga, J. D., Tarquino, J. and Romero, E., "Characterizing Architectural Distortion in Mammograms by Linear Saliency," J. Med. Syst. **41**(2), 26 (2017).
[34] Wang, S., Rao, R. V., Chen, P., Zhang, Y., Liu, A. and Wei, L., "Abnormal Breast Detection in Mammogram Images by Feed-forward Neural Network Trained by Jaya Algorithm," Fundam. Informaticae **151**(1–4), 191–211 (2017).
[35] Nithya, R. and Santhi, B., "Classification of normal and abnormal patterns in digital mammograms for diagnosis of breast cancer," Int. J. Comput. Appl. **28**(6), 21–25 (2011).
[36] Zhu, W., Lou, Q., Vang, Y. S. and Xie, X., "Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification," ArXiv161205968 Cs (2016).
[37] Jiao, Z., Gao, X., Wang, Y. and Li, J., "A deep feature based framework for breast masses classification," Neurocomputing **197**, 221–231 (2016).
[38] Dhungel, N., Carneiro, G. and Bradley, A. P., "The Automated Learning of Deep Features for Breast Mass Classification from Mammograms," Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2016, 106–114, Springer, Cham (2016).
[39] Borges Sampaio, W., Moraes Diniz, E., Corrêa Silva, A., Cardoso de Paiva, A. and Gattass, M., "Detection of masses in mammogram images using CNN, geostatistic functions and SVM," Comput. Biol. Med. **41**(8), 653–664 (2011).
[40] Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," ArXiv14091556 Cs (2014).
[41] Friedewald, S. M., Rafferty, E. A., Rose, S. L., Durand, M. A., Plecha, D. M., Greenberg, J. S., Hayes, M. K., Copit, D. S., Carlson, K. L., Cink, T. M., Barke, L. D., Greer, L. N., Miller, D. P. and Conant, E. F., "Breast Cancer Screening Using Tomosynthesis in Combination With Digital Mammography," JAMA **311**(24), 2499–2507 (2014).
[42] Sharma, A., [DDSM Utility], GitHub (2015).
[43] Nair, V. and Hinton, G. E., "Rectified linear units improve restricted boltzmann machines," Proc. 27th Int. Conf. Mach. Learn. ICML-10, 807–814 (2010).
[44] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting.," J. Mach. Learn. Res. **15**(1), 1929–1958 (2014).
[45] Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization," ArXiv14126980 Cs (2014).
[46] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," ArXiv160304467 Cs (2016).
[47] Radford, A., Metz, L. and Chintala, S., "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," ArXiv151106434 Cs (2015).
[48] Hindupur, A., [the-gan-zoo: A list of all named GANs!] (2018).
[49] Zhu, W., Xiang, X., Tran, T. D., Hager, G. D. and Xie, X., "Adversarial Deep Structured Nets for Mass Segmentation from Mammograms," ArXiv171009288 Cs (2017).
[50] Rezaei, M., Harmuth, K., Gierke, W., Kellermeier, T., Fischer, M., Yang, H. and Meinel, C., "Conditional Adversarial Network for Semantic Segmentation of Brain Tumor," ArXiv170805227 Cs (2017).

[51] Son, J., Park, S. J. and Jung, K.-H., "Retinal Vessel Segmentation in Fundoscopic Images with Generative Adversarial Networks," ArXiv170609318 Cs (2017).
[52] Kohl, S., Bonekamp, D., Schlemmer, H.-P., Yaqubi, K., Hohenfellner, M., Hadaschik, B., Radtke, J.-P. and Maier-Hein, K., "Adversarial Networks for the Detection of Aggressive Prostate Cancer," ArXiv170208014 Cs (2017).
[53] Hu, Y., Gibson, E., Lee, L.-L., Xie, W., Barratt, D. C., Vercauteren, T. and Noble, J. A., "Freehand Ultrasound Image Simulation with Spatially-Conditioned Generative Adversarial Networks," [Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment], Springer, Cham, 105–115 (2017).
[54] Chuquicusma, M. J. M., Hussein, S., Burt, J. and Bagci, U., "How to Fool Radiologists with Generative Adversarial Networks? A Visual Turing Test for Lung Cancer Diagnosis," ArXiv171009762 Cs Q-Bio (2017).
[55] Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q. and Shen, D., "Medical Image Synthesis with Context-Aware Generative Adversarial Networks," Med. Image Comput. Comput.-Assist. Interv. − MICCAI 2017, 417–425, Springer, Cham (2017).
[56] Bi, L., Kim, J., Kumar, A., Feng, D. and Fulham, M., "Synthesis of Positron Emission Tomography (PET) Images via Multi-channel Generative Adversarial Networks (GANs)," [Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment], Springer, Cham, 43–51 (2017).
[57] Guibas, J. T., Virdi, T. S. and Li, P. S., "Synthetic Medical Images from Dual Generative Adversarial Networks," ArXiv170901872 Cs (2017).
[58] Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J. and Ré, C., "Learning to Compose Domain-Specific Transformations for Data Augmentation," [Advances in Neural Information Processing Systems 30], I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 3239–3249 (2017).
[59] Ren, S., He, K., Girshick, R. and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," ArXiv150601497 Cs (2015).
[60] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., "Rethinking the Inception Architecture for Computer Vision," ArXiv151200567 Cs (2015).
[61] He, K., Zhang, X., Ren, S. and Sun, J., "Deep Residual Learning for Image Recognition," ArXiv151203385 Cs (2015).
[62] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A., "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," ArXiv160207261 Cs (2016).