

COVID-19 diagnosis on CT scan images using a generative adversarial network and concatenated feature pyramid network with an attention mechanism

Zonggui Li | Junhua Zhang | Bo Li | Xiaoying Gu | Xudong Luo

School of Information Science and Engineering, Yunnan University, Kunming, China

Correspondence

Junhua Zhang, School of Information Science and Engineering, Yunnan University, East Outer Ring Road, Chengong District, Kunming 650500, China.
Email: jhzhzhang@ynu.edu.cn

Funding information

National Natural Science Foundation of China (NSFC), Grant/Award Number: 62063034

Abstract

Objective: Coronavirus disease 2019 (COVID-19) has caused hundreds of thousands of infections and deaths. Efficient diagnostic methods could help curb its global spread. The purpose of this study was to develop and evaluate a method for accurately diagnosing COVID-19 based on computed tomography (CT) scans in real time.

Methods: We propose an architecture named “concatenated feature pyramid network” (“Concat-FPN”) with an attention mechanism, by concatenating feature maps of multiple. The proposed architecture is then used to form two networks, which we call COVID-CT-GAN and COVID-CT-DenseNet, the former for data augmentation and the latter for data classification.

Results: The proposed method is evaluated on 3 different numbers of magnitude of COVID-19 CT datasets. Compared with the method without GANs for data augmentation or the original network auxiliary classifier generative adversarial network, COVID-CT-GAN increases the accuracy by 2% to 3%, the recall by 2% to 4%, the precision by 1% to 3%, the F1-score by 1% to 3%, and the area under the curve by 1% to 4%. Compared with the original network DenseNet-201, COVID-CT-DenseNet increases the accuracy by 1% to 3%, the recall by 4% to 9%, the precision by 1%, the F1-score by 1% to 3%, and the area under the curve by 2%.

Conclusion: The experimental results show that our method improves the efficiency of diagnosing COVID-19 on CT images, and helps overcome the problem of limited training data when using deep learning methods to diagnose COVID-19.

Significance: Our method can help clinicians build deep learning models using their private datasets to achieve automatic diagnosis of COVID-19 with a high precision.

KEYWORDS

attention mechanism, concatenated feature pyramid network, COVID-19, CT images, generative adversarial network

1 | INTRODUCTION

According to data released by the World Health Organization in November 2020, there have been more than 57.8 million confirmed cases of coronavirus disease 2019 (COVID-19) worldwide, with nearly 1.3 million deaths.¹ Speeding up the diagnosis of COVID-19 is of great significance for saving millions of lives, and restoring the economic development of countries. The current tests commonly used to diagnose COVID-19 include reverse transcription-polymerase chain reaction (RT-PCR), chest X-rays, and computed tomography (CT). The average turnaround time for a RT-PCR test is 24 h, which greatly slows the testing process. Moreover, the test kits are limited in supply and are expensive. An investigation 1014 cases found that chest CT is highly sensitive to the diagnosis of COVID-19, and the detection speed is faster than RT-PCR.² In addition, the report from Fang et al.³ showed that the sensitivity of chest CT was greater than that of RT-PCR (98% versus 71%, respectively). However, as a new disease, COVID-19 has similar manifestations

with other types of pneumonia. Therefore, analyzing and diagnosing the characteristics of COVID-19 in CT images relies heavily on clinical expertise, and is time-consuming.⁴ As shown in Figure 1, the most commonly reported CT finding in COVID-19 is that infection can cause severe lower respiratory tract infection, accompanied by basal, bilateral, and peripheral dominant ground-glass opacity, which is a typical feature of lung tissue damage from pneumonia.⁵ Diffuse airspace opacity and asymmetric patchiness were also reported in patients with COVID-19.⁶ Such subtle abnormalities are difficult for clinicians, except expert radiologists, to interpret. Considering the limited number of professionally trained radiologists, using automatic methods such as deep learning⁷ to identify such subtle abnormalities could assist in the diagnostic process and improve the early diagnosis rate.

Research in recent years has proven that deep learning can effectively diagnose diseases from medical images.⁸ Many recent studies have, therefore, used deep learning methods to analyze CT scan images to assist in the diagnosis of COVID-19, and some of them have

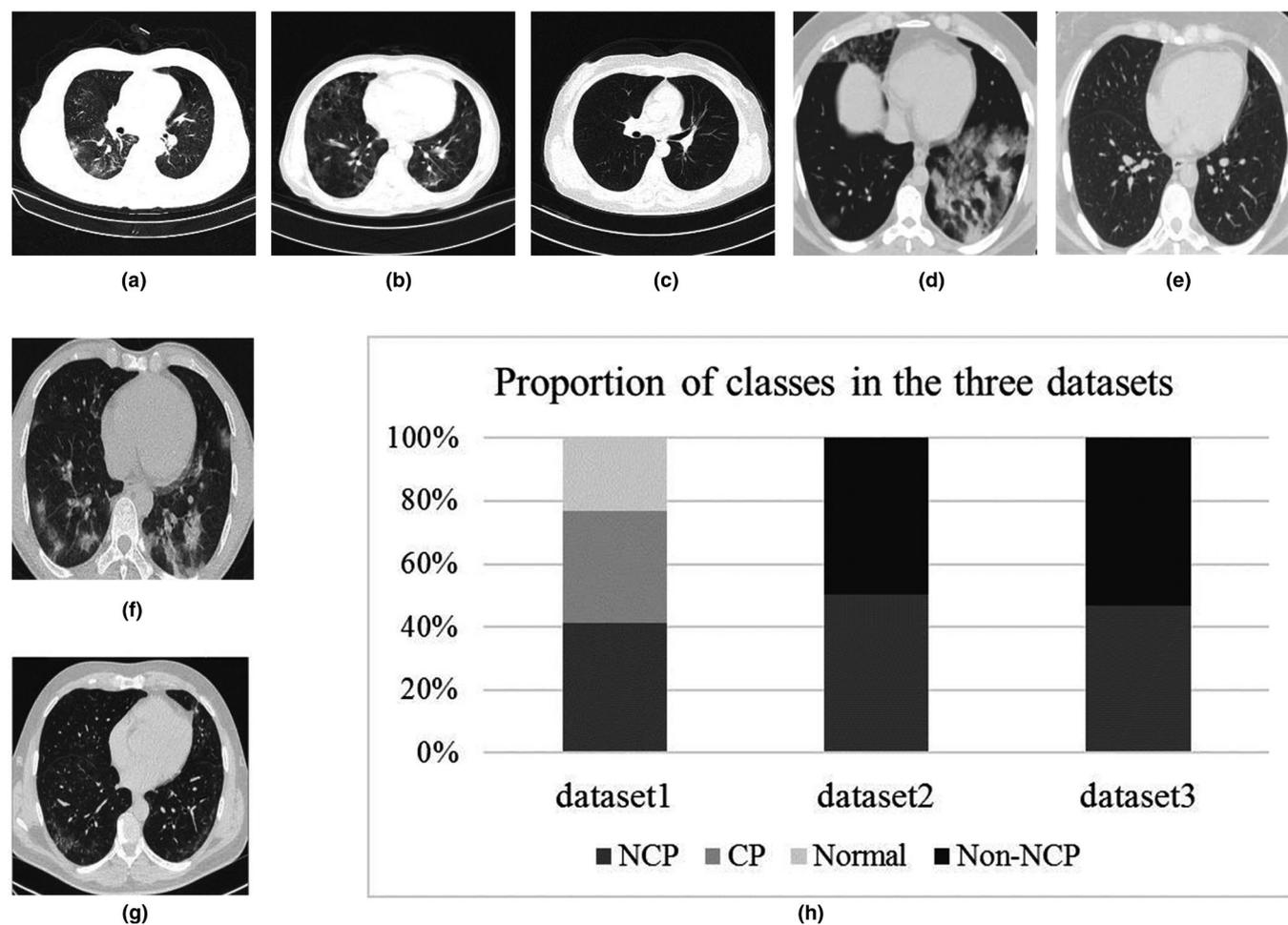


FIGURE 1 Examples of CT images. (a–c) Novel coronavirus pneumonia (NCP), common pneumonia (CP), and normal controls (Normal) from dataset 1, respectively. (d and e) NCP and Non-NCP from dataset 2, respectively. (f and g) NCP and Non-NCP from dataset 3, respectively. (h) The proportion of classes in the datasets

published their datasets. The SARS-CoV-2 CT-scan dataset⁹ is a publicly available dataset from COVID-19 containing 1252 CT scans from patients infected with SARS-CoV-2 and 1230 from non-infected patients. By using an explainable deep learning approach (xDNN), the baseline result for that dataset is 0.973(F1 score). He et al.¹⁰ assembled a COVID-19 CT dataset that contains 349 positive and 397 negative CT scans. They also proposed a self-supervised transfer learning approach based on a deep learning model, such as DenseNet-169¹¹ to achieve an F1-score of 0.850. It is commendable that He et al.¹² compiled a new dataset containing a total of 340,190 slices of 3993 scans from 2698 patients, from a large COVID-19 CT public dataset called CC-CCII.¹³ They also used a series of state-of-the-art three-dimensional (3D) convolutional neural network (CNN) models (such as DenseNet3D-121¹⁴) to achieve a good classification effect with an F1-score of 0.872. Coincidentally, Ouyang et al.¹⁵ also used a 3D CNN model to diagnose COVID-19 on CT scan images. They used two different sampling strategies to train two 3D ResNet34 networks¹⁴ with an attention mechanism, and finally integrated predictions from the two networks using ensemble learning. There have been some reports regarding diagnosing COVID-19 based on the infected regions in CT images. Wang et al.¹⁶ collected 453 CT scan images of COVID-19 and used the inception migration-learning model to distinguish those cases from other cases of viral pneumonia, based on manually labeled regions. Similarly, Xu et al.¹⁷ segmented the infected regions in scans from COVID-19 using a V-Net model and classified the images using a ResNet-18 network. Matsuyama et al.¹⁸ proposed a ResNet-50-based CNN model to discriminate COVID-19 from Non-COVID-19 (No COVID-19 infection) using chest CT and used a gradient-weighted activation map to interpret the model visually. In addition, Song et al.¹⁹ extracted images with the relevant lung regions from slices of each group of processed 3D CT images, and they used ResNet-50 to form a prediction for each CT image. Kang et al.²⁰ proposed the use of multi-view representation learning to diagnose COVID-19. There was also a study tried to optimize the training process of the CNN model to improve the classification accuracy: Anwar et al.²¹ achieved an F1 score of 0.896 using EfficientNet deep learning architecture with three different learning rates strategies (i.e., cyclic learning rate, reducing the learning rate when model performance stops increasing, and constant learning rate).

So far, researchers can only train deep learning models using limited CT scan images to perform automatic COVID-19 diagnosis, because of the lack of publicly available large-sample CT scan image datasets from COVID-19 patients. As the most commonly used deep learning model, CNN can attain state-of-the-art performance in medical imaging if sufficient data are

available.^{8,22} Such performance can be achieved by training on labeled data with its millions of parameters. However, due to the large number of training parameters, CNN is easy to overfit on small datasets. This phenomenon is especially obvious in medical image datasets that require radiologists to participate in labeling.^{23,24} Therefore, researchers usually use data augmentation methods to expand datasets. The specific method involves making some simple modifications to the dataset images, such as rotating, translating, scaling, and flipping. It is a standard procedure to use these classic data augmentation methods in the field of computer vision to improve the training process of the network.²⁵ Based on the above analysis, to solve the problem of limited samples in the COVID-19 CT scan image dataset, we propose to use an improved generative adversarial network (GAN),²⁶ which we called COVID-CT-GAN for data augmentation. The COVID-CT-GAN we designed is based on an auxiliary classifier GAN (ACGAN).²⁷ To the best of our knowledge, this is the first attempt to augment COVID-19 CT image data using a GAN, and to explore the impact of the network architecture on the classification of the images generated. The main contributions of our work are summarized as follows:

- We propose an optimized GAN to replace the traditional methods to augment COVID-19 CT scan image training data.
- We propose a concatenated feature pyramid network (Concat-FPN) with an attention mechanism, and apply it to a GAN and a DenseNet-201 model for generating and classifying COVID-19 CT scan images, respectively.
- We explore and summarize the impact of the number of upsampling layers of the generator and the size of the convolutional kernels on the performance of our proposed COVID-CT-GAN.
- We evaluate our methods on three COVID-19 CT scan datasets of different numbers of images (magnitudes of 1×10^2 , 1×10^3 , and 1×10^4) and summarize the impact of dataset size on the use of a GAN for data augmentation.

In the rest of this paper, Sections 2 and 3 review related works and detail our method, respectively. Section 4 presents the experiments and discussion. Section 5 presents the conclusions.

2 | RELATED WORKS

2.1 | Generative adversarial network

Since GANs were first proposed, they have been widely used in medical imaging.²⁸ According to the characteristics of GANs composed of generator and

discriminator, they are primarily used in the following two ways in medical imaging. The first way is to use the generator to learn the feature distribution of the training data and generate new images, which helps solve the problem of insufficient samples in the medical image dataset. CT image denoising,²⁹ transformation between brain CT images and magnetic resonance images,^{30,31} retinal image synthesis,³² and generation of dermoscopic images melanoma³³ are some examples of generator-related applications. The second approach is to use the discriminator as a detector to distinguish abnormal images. One such example entails the detection of anomalies in optical coherence tomography images of the retina.³⁴

In the research into medical image classification, the generator and discriminator of GANs are usually used for image feature extraction, or the discriminator is directly used as a classifier by adding additional categories. Hu et al.³⁵ unified InfoGAN³⁶ and WGAN³⁷ through the loss function for unsupervised feature learning in histopathology images, whereas Yi et al.³³ combined CatGAN³⁸ and WGAN for semi-supervised and unsupervised feature learning in dermoscopy images. Both studies used the features extracted by the discriminator for classification. Moreover, a semi-supervised GAN also shows an advantage, compared with traditional supervised CNNs, by achieving comparable performance with fewer labeled data. For example, Lahiri et al.³⁹ and Madani et al.⁴⁰ used the semi-supervised deep convolutional generative adversarial network (DCGAN)⁴¹ for retinal vessel classification and cardiovascular abnormality classification in chest X-rays, respectively. Madani et al.⁴² also used two DCGANs to generate normal and abnormal chest X-rays separately, which achieved higher classification accuracy than traditional data augmentation methods. This may be due to the fact that traditional data augmentation methods cannot capture the biological variance of medical images, which may result in unrealistic images after augmentation. In addition, generating each class separately (i.e., N models for N classes) may lead to low computational efficiency. Therefore, a potential research direction is the use of a single model for multi-class conditional synthesis.⁴³ However, there are some special cases. For example, when Frid-Adar et al.⁴⁴ used a GAN to synthesize liver lesions for classification, they found that the use of a separated GAN (DCGAN) produced better results than did a unified GAN (ACGAN). The reason may be the specificity of the dataset, but the deeper reasons need to be explored. Further, some researchers argue that the images generated by GANs may be effective for augmentation in the low-data regime, but may not be helpful in the high-data regime.⁴⁵⁻⁴⁸ Coincidentally, Karadağ et al.⁴⁹ found that the performance of GANs for data augmentation is closely related to model hyperparameters (e.g., learning rate and number of epochs) and the size of the dataset in the image classification problem.

Aiming at the use of GANs for data augmentation, we conducted experimental exploration of three COVID-19 CT image datasets.

2.2 | Layer fusion and attention mechanism

Even if CNNs and GANs can learn the differences between different types of COVID-19 CT scan images, there are still some discrepancies due to the differences in features between different scans. In many works, fusing features of different scales is an important way to improve the performance of convolutional networks. Between these features, the low-level feature semantic information is limited, but the target location is accurate; the high-level feature semantic information is rich, but the target location is relatively coarse. According to the order of fusion and prediction, feature fusion is divided into early fusion and late fusion. Inside-Outside Net (ION)⁵⁰ and HyperNet⁵¹ are examples of early fusion, whereas Single Shot MultiBox Detector (SSD),⁵² Multi-scale CNN (MS-CNN)⁵³ and feature pyramid network (FPN)⁵⁴ are examples of late fusion. These methods are mainly used for object detection in images, specifically to add and concatenate, to connect different levels of feature extraction layers. Recent work has also shown that the connection of CNNs and the connection of GANs can effectively improve their performance. Huang et al.¹¹ proposed the densely connected convolutional network (DenseNet), which concatenated the input layers with the output layers iteratively, and achieved significant improvements on four large classification datasets. Wang et al.⁵⁵ proposed a concatenated 3D conditional GAN to improve the quality of generated images when extracting full-dose positron emission tomography images from low-dose ones. It is worth mentioning that the feature maps of two adjacent layers are resized to the same size through upsampling in the original FPN, which is generally used for object detection, and the two layers are merged by addition. We conducted comparative experiments and found that merging the two adjacent layers by concatenation can achieve better results. Therefore, we design a concatenated FPN that we call Concat-FPN, and applied it to GAN and DenseNet for COVID-19 CT images generation and classification.

TABLE 1 Statistics of the datasets

Dataset	Patients	Slices	NCP	Non-NCP	
				CP	Normal
①	2729	135,609	56,198	47,745	31,666
②	120	2482	1252	1230	
③	143	746	349	397	

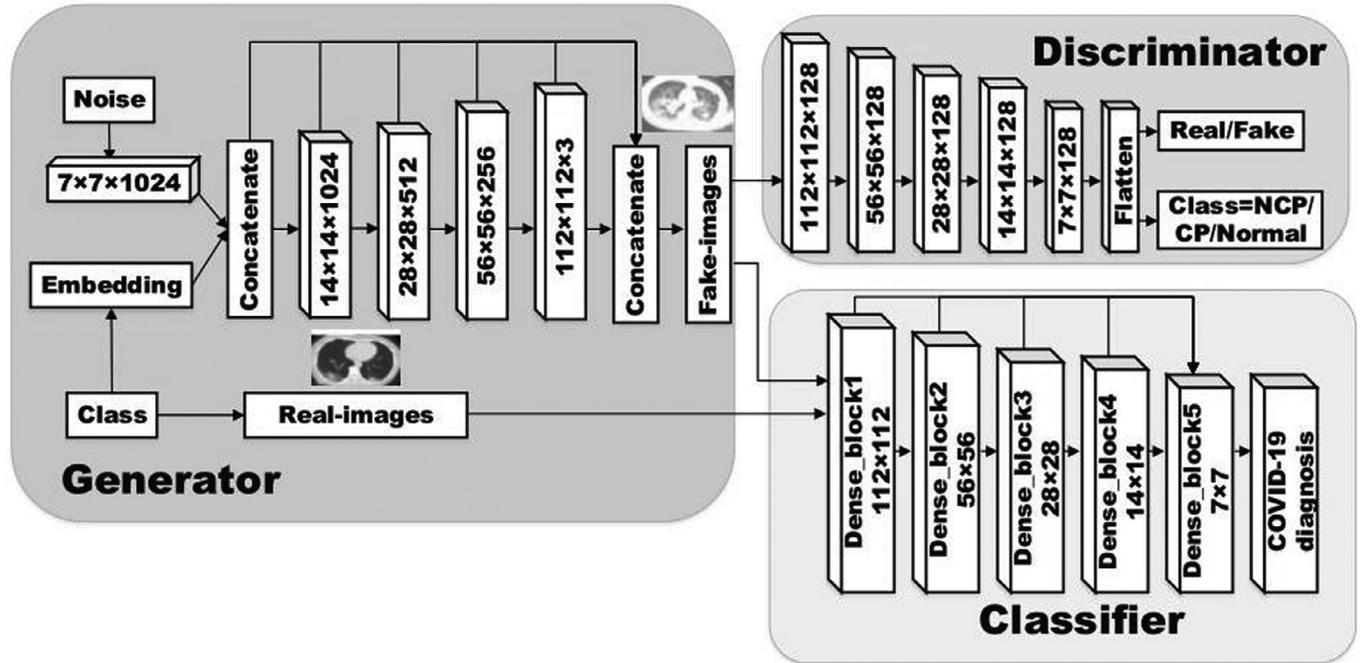


FIGURE 2 Overview of our proposed framework

To better capture the structure of objects, the human visual system will focus on salient parts selectively.⁵⁶ In view of this, an attention mechanism was then developed based upon human perception and applied to computer vision tasks, achieving great results.^{57–59} Recently, researchers have used attention mechanisms to improve the classification performance of CNNs in several large-scale image classification tasks.^{60–62} Unlike the way in which Woo et al.⁶² integrated the proposed Convolutional Block Attention Module (CBAM) with the ResBlock in ResNet, in this study, we integrated CBAM with our Concat-feature pyramid network.

3 | MATERIALS AND METHODS

3.1 | Dataset

We evaluated the proposed method using three different COVID-19 CT scan datasets, all of which are downloadable. The statistics of the datasets are shown in Table 1. We aimed to diagnose COVID-19 from a single CT scan image, using deep learning methods. The dataset published by He et al.¹² and its original dataset CC-CCII¹³ contain a large number of COVID-19 CT scan images. However, there are many CT images in the datasets that do not show the lesion, and even some images in which the lungs are completely invisible. We believe that this would have had a negative impact on the classification effect, that is, it would mean that the neural network training process could not learn the lesion characteristics of COVID-19 very well.

Therefore, we excluded the images that did not contain lungs or lesions, and constructed dataset 1 based on these two large datasets. Dataset 1 contained three classes (i.e., novel coronavirus pneumonia (NCP), normal controls (Normal), and common pneumonia (CP)). Datasets 2 and 3 come from the work of Soares et al.⁹ and He et al.,¹⁰ respectively. Both of those datasets contained only positive and negative CT scan images of COVID-19 (i.e., NCP and Non-NCP). For convenience, we denote datasets 1, 2, and 3 as,, and, respectively. In our experiment, we divided each dataset into 5 pieces. In each piece of data, the proportion of the classes of the CT image was consistent with that in the original dataset. In addition, the proportion of the training set, validation set, and test set was 3:1:1. The CT image data was transformed into an array, and then fed into the neural network we designed. The implemented code and the three datasets are downloadable*.

3.2 | Framework

The overall framework we propose is shown in Figure 2. It contains two stages: (1) based on ACGAN,²⁷ we built a generator that can generate two or three classes of lung CT scan images (i.e., NCP, CP, and Normal) for data augmentation, and (2) we constructed a classifier for diagnosing COVID-19 based on DenseNet-201. In both stages, we used the Concat-FPN with the attention mechanism we proposed. The following sections detail our method.

*<https://github.com/lizonggui/COVID-19>

3.3 | Concat-feature pyramid network with attention mechanism

As shown in Figure 3, the original FPN was merged by addition, using lateral connections and top-down pathways. The position of the object on multiple scales could be predicted in the FPN. Starting from the last layer, upsampling the output feature map of the latter layer by a factor of 2 (using nearest neighbor upsampling) was

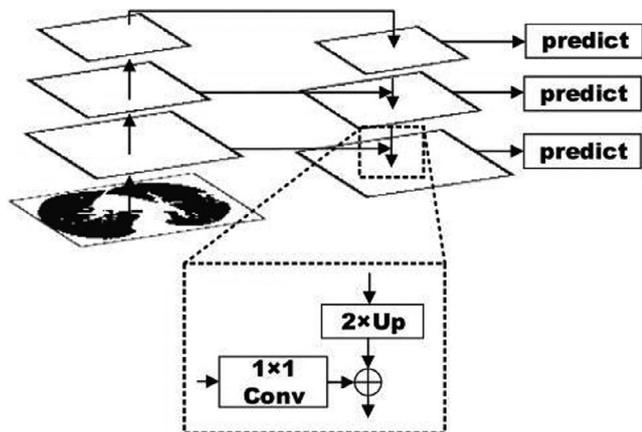


FIGURE 3 Traditional FPN architecture, merged by addition

performed, and then that output was merged by addition with the output feature map of the former layer, which underwent a 1×1 convolutional layer. In this study, we attempted to merge the two feature maps by concatenation, instead of addition. Concatenation is an important operation in the design of network structure. It is often used to combine features, fuse features extracted by multiple convolution feature extraction frameworks, or fuse the information of the output layers, whereas adding layers is more like the superposition of information. The network structure we built, called Concat-FPN, can develop an output on the final fusion feature map. Due to the structural difference between the classifier network and the generator of the GAN, the structure of Concat-FPN is slightly different when used in these two, and we describe them separately.

Consider y_l is the output of the l^{th} layer. $Conv$ is the convolution operation. Up is the upsampling operation. \otimes represents the concatenation. For the classifier network, the output of Concat-FPN with k layers is

$$Y = Conv_{s=2^0, 3 \times 3} [Conv_{s=1, 1 \times 1} (y_{k-1}) \otimes Up_{2 \times} (y_k)] \otimes Conv_{s=2^1, 3 \times 3} [Conv_{s=1, 1 \times 1} (y_{k-2}) \otimes Up_{2 \times} (y_{k-1})] \otimes \dots \otimes Conv_{s=2^{k-2}, 3 \times 3} [Conv_{s=1, 1 \times 1} (y_1) \otimes Up_{2 \times} (y_2)], \quad (1)$$

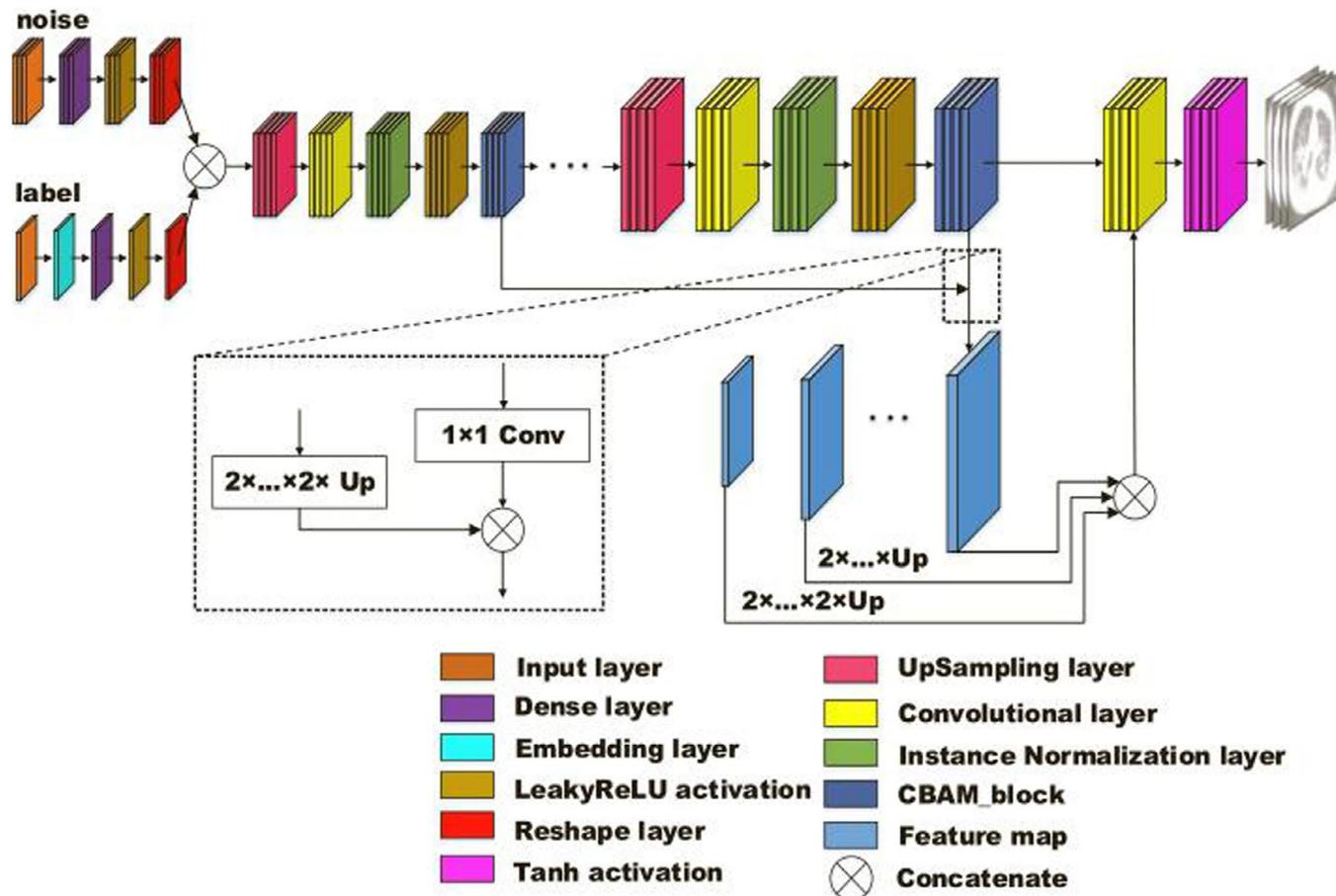


FIGURE 4 Layered architecture of COVID-CT-GAN generator

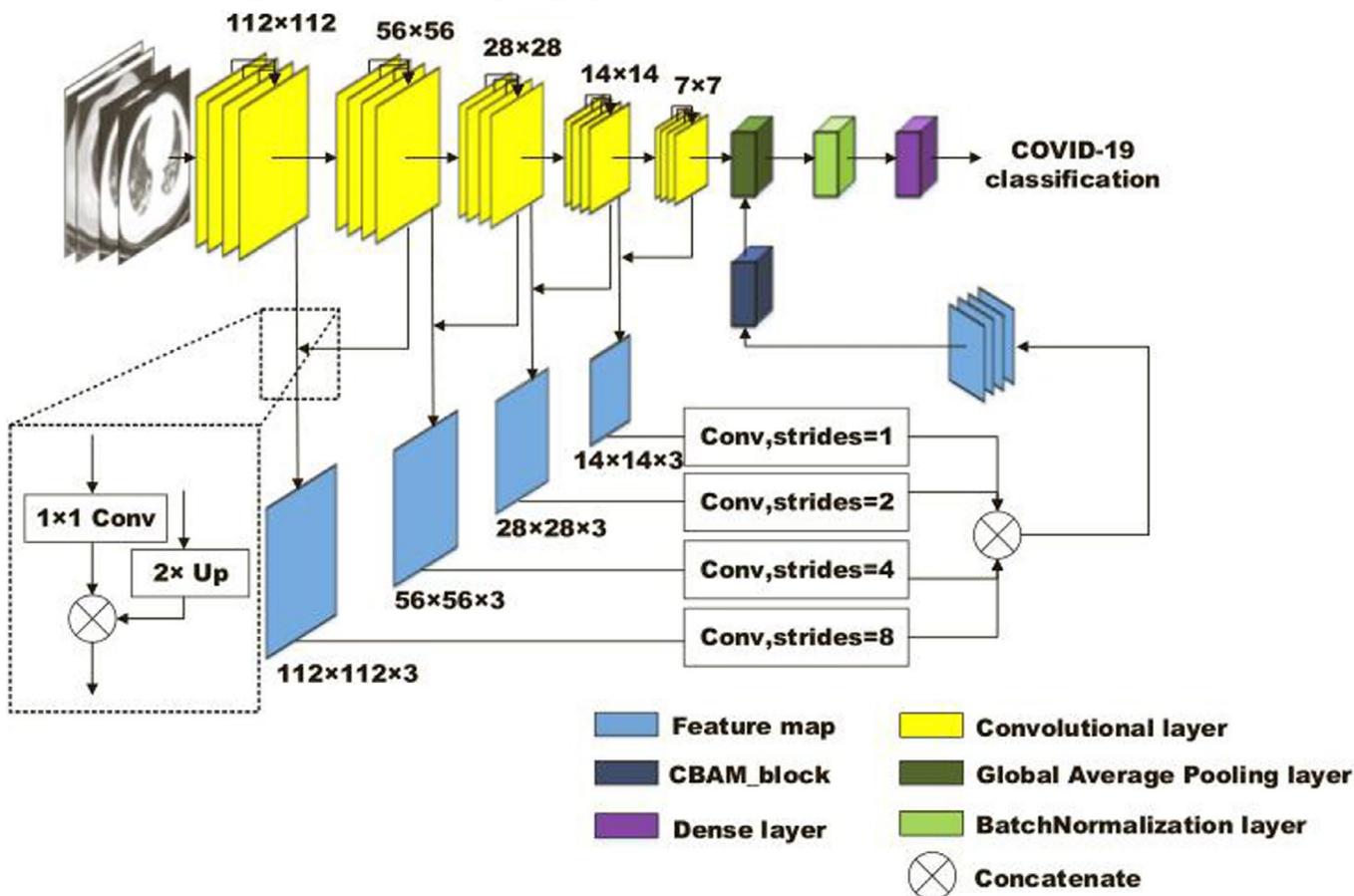


FIGURE 5 The structure of classifier network

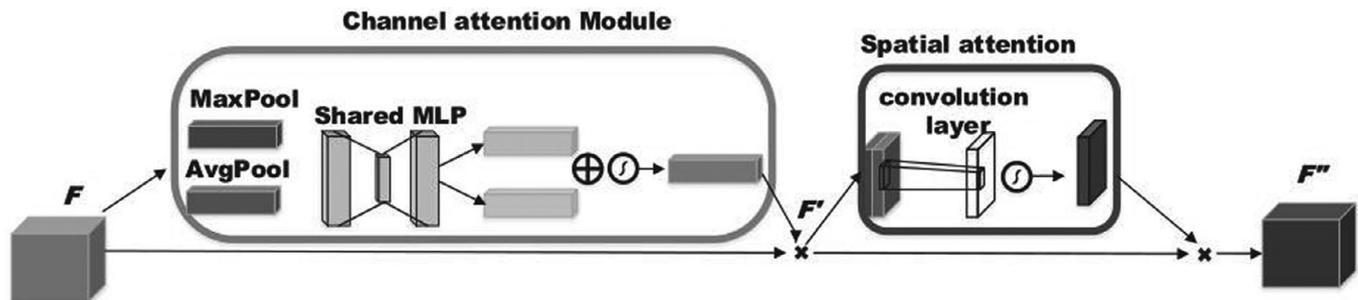


FIGURE 6 The structure of convolutional block attention module

where s is the stride of the convolution, 1×1 and 3×3 are the sizes of the convolution kernels, and $2 \times$ denotes an upsampling factor of 2. It is worth noting that the 3×3 convolution on each merged map serves two purposes. One is to reduce the aliasing effect caused by upsampling, and the other is to resize feature maps of different sizes, to be the same size for concatenation.

For the generator of the GAN, in contrast to the classifier network, this is an upsampling process. To fuse feature maps of different sizes, we upsampled the previous feature maps to be the same size, first, and then

performed fusion. As shown in Figure 4, the output of Concat-FPN with k layers is

$$Y = [Up_{2 \times}(y_{k-1}) \otimes Conv_{s=1,1 \times 1}(y_k) \otimes [Up_{2^2 \times}(y_{k-2}) \otimes Conv_{s=1,1 \times 1}(y_k)] \otimes \dots \otimes [Up_{2^{k-1} \times}(y_1) \otimes Conv_{s=1,1 \times 1}(y_k)], \quad (2)$$

where $Up_{2^{k-1} \times}$ denotes an upsampling factor of 2^{k-1} .

Furthermore, we argue that resizing feature maps of different sizes through FPN, and then directly fusing them, may be slightly coarse. Merging the salient parts

TABLE 2 Training parameters

Networks	Optimizer	Learning rate	Weight decay	Epoch	Batch size
COVID-CT-DenseNet	Adam	1e-4 (reduced with 0.8)	0.0001	50	32
COVID-CT-GAN	Adam	1e-4	0.0001	40 000	4

TABLE 3 The results of different classifier networks

Dataset	Networks	Accuracy	Recall	Precision	F1-score	AUC
①	VGG-19	0.83	0.78	0.84	0.80	—
	EfficientNet-B5	0.83	0.77	0.88	0.82	—
	ResNet-101	0.80	0.74	0.86	0.80	—
	ResNeXt-101	0.85	0.79	0.88	0.83	—
	DenseNet-169	0.85	0.79	0.90	0.84	—
	DenseNet-201	0.82	0.76	0.87	0.81	—
	COVID-CT-DenseNet	0.85	0.80	0.88	0.84	—
②	VGG-19	0.84	0.77	0.91	0.83	0.90
	EfficientNet-B5	0.83	0.69	0.96	0.80	0.95
	ResNet-101	0.82	0.77	0.87	0.82	0.90
	ResNeXt-101	0.81	0.66	0.95	0.78	0.89
	DenseNet-169	0.86	0.74	0.99	0.85	0.97
	DenseNet-201	0.90	0.83	0.98	0.90	0.97
	COVID-CT-DenseNet	0.91	0.92	0.91	0.91	0.97
③	VGG-19	0.72	0.81	0.68	0.73	0.81
	EfficientNet-B5	0.75	0.81	0.72	0.76	0.83
	ResNet-101	0.77	0.76	0.76	0.76	0.84
	ResNeXt-101	0.76	0.84	0.72	0.77	0.86
	DenseNet-169	0.77	0.83	0.74	0.78	0.87
	DenseNet-201	0.79	0.79	0.79	0.79	0.88
	COVID-CT-DenseNet	0.82	0.85	0.80	0.82	0.90

of the feature maps may improve the performance of the network. Therefore, we attempted to use an attention mechanism in the process of feature fusion. Specifically, we used Concat-FPN in the generator of the GAN to merge attention maps of different sizes, as illustrated in Figure 4, whereas in the classifier network, Concat-FPN was used to merge all feature maps and then its attention map was sent to the classification layer, as illustrated in Figure 5. In the network, we used CBAM⁶² to generate attention maps. The structure of the CBAM is shown in Figure 6. It is composed of a channel attention module and a spatial attention module. Given a feature map F , its attention map generated by CBAM is

$$F'' = [\sigma(W_1(W_0(\text{AvgPool}(F))) + W_1(W_0(\text{MaxPool}(F))))] * [\sigma(\text{Conv}_{7 \times 7}([\text{AvgPool}(F); \text{MaxPool}(F)])], \quad (3)$$

where *AvgPool* and *MaxPool* denote the average-pooling operation and max-pooling operation, respectively. σ is the sigmoid function. $*$ is the element-wise multiplication. W_1 and W_0 are the weights of the shared network, which is a multi-layer perceptron with one dense layer. The size of the convolutional kernel is 7×7 .

3.4 | COVID-CT-GAN

As shown in Figures 2 and 4, the generator of COVID-CT-GAN that we designed took a vector of random noise from latent space and the category label as input. The two inputs were concatenated and passed through one to four upsampling blocks. We explored the impact of the number of upsampling layers and the size of convolutional kernels on the performance of

MEDICAL PHYSICS

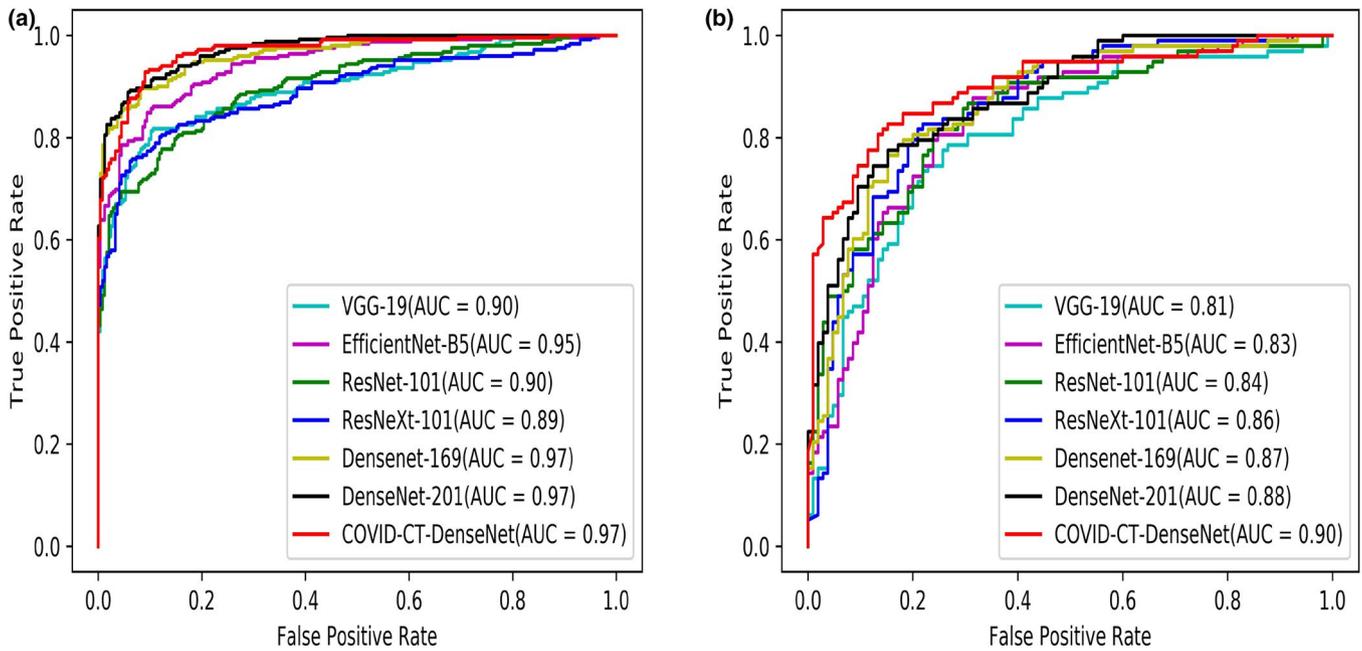


FIGURE 7 The ROC curves of different classifier networks on ② and ③. The horizontal axis represents the false positive rate, and the vertical axis represents the true positive rate [Color figure can be viewed at wileyonlinelibrary.com]

Dataset	Networks	Accuracy	Recall	Precision	F1-score	AUC
①	None	0.85	0.80	0.88	0.84	—
	ACGAN	0.84	0.81	0.86	0.83	—
	DCGAN	0.79	0.75	0.77	0.76	—
	COVID-CT-GAN	0.84	0.79	0.89	0.84	—
②	None	0.91	0.92	0.91	0.91	0.97
	ACGAN	0.92	0.94	0.93	0.93	0.96
	DCGAN	0.85	0.87	0.85	0.86	0.94
	COVID-CT-GAN	0.93	0.96	0.93	0.94	0.98
③	None	0.82	0.85	0.80	0.82	0.90
	ACGAN	0.84	0.85	0.81	0.83	0.92
	DCGAN	0.81	0.80	0.79	0.79	0.88
	COVID-CT-GAN	0.85	0.87	0.83	0.85	0.94

TABLE 4 The results of GANS for data augmentation.

the GAN through experiments. The upsampling blocks consisted of the upsampling layer, convolutional layer, instance normalization layer, leaky rectified linear units (ReLU) activation layer and the CBAM. The pool-size of each upsampling layer was 2×2 . The numbers of filters in the convolutional layer were 1024, 512, 256, and 128 (corresponding to upsampling block numbers of one, two, three, and four, respectively). At the end of the network, a convolutional layer with three channels and a tanh activation layer were applied to the output. As mentioned before, we used Concat-FPN to concatenate the attention maps of different upsampling blocks, and then sent the concatenated map to the output.

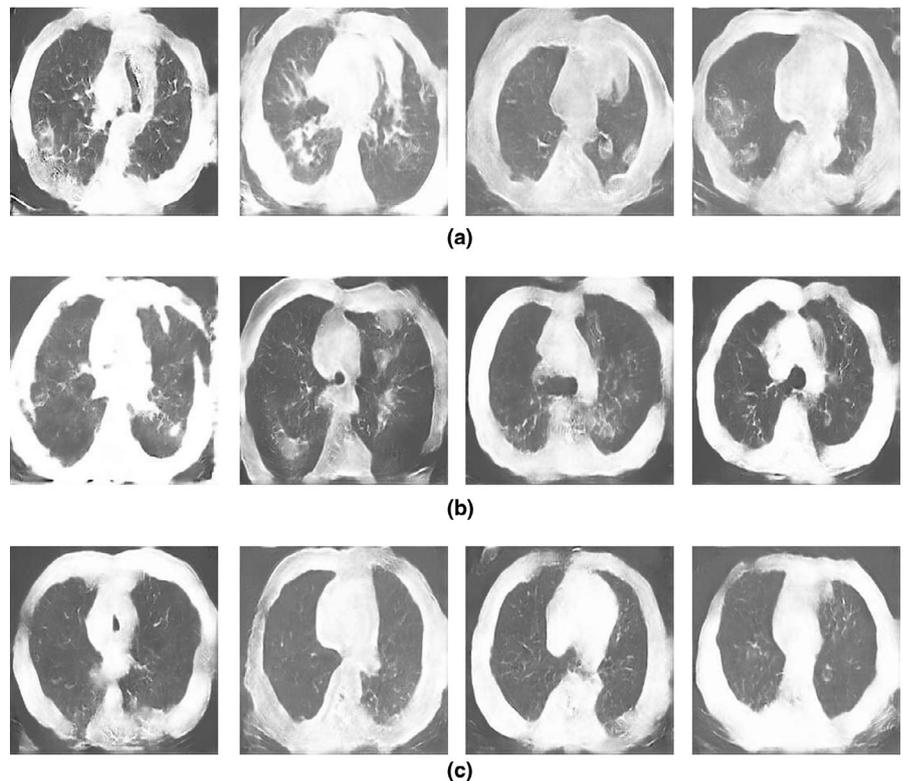
The discriminator was a structure of down-sampling as shown in Figure 2. To train COVID-CT-GAN stably,

the number of filters was set to 128, and the stride was set to 2 in each convolutional layer. The leaky ReLU activation layer was used after each convolutional layer. The output of the last convolutional layer was flattened, followed by two dense layers as classifiers. The discriminator had two main functions. One was to classify the images produced by the generator. The other was to promote the generator to produce images closer to the real ones in the process of adversarial training.

3.5 | COVID-CT-DenseNet

The original data and that generated by COVID-CT-GAN were used to train the classifier network to

FIGURE 8 Examples of the three classes of CT images generated by COVID-CT-GAN. (a) NCP; (b) CP; (c) Normal



diagnose COVID-19. As shown in Figure 5, we used the DenseNet-201 architecture¹¹ as the backbone network. It has five dense blocks, which connect each layer to each other layer in a feed-forward fashion. We note that the concatenation operation was completed in each dense block separately, because the operation can only be used on feature maps of the same size. We argue that concatenating the feature maps of these five dense blocks at the same time may improve network performance. Therefore, we used Concat-FPN to fuse the output feature maps that had been concatenated separately in these five dense blocks. To highlight the extracted features, the fused output map was fed into the CBAM block, to generate the attention map, and then was input into a global average pooling layer and a batch normalization layer, to prevent overfitting. Finally, a dense layer with SoftMax activation function was used to output the classification result to complete the diagnosis of COVID-19. We call the designed classifier network COVID-CT-DenseNet.

4 | RESULTS

4.1 | Experimental settings and evaluation measures

We implemented the network architecture of COVID-CT-DenseNet and COVID-CT-GAN in Keras.⁶³ All the models were trained using NVIDIA GeForce RTX

2080Ti GPU with 11GB memory and Intel Core i7 9700K CPU with 32-GB RAM. The training parameter settings are shown in Table 2.

We used the following five metrics to evaluate our approaches: (1) Accuracy, which represents the proportion of samples that the diagnostic predictions matched with the truth. (2) Recall, which is the proportion of all the true-positive samples that were predicted to be positive. (3) Precision, which is the proportion of all the samples with positive predictions that were true positives. (4) F1-score, which is defined as the harmonic mean of recall and precision. (5) AUC, which is the area under the receiver operating characteristic curve (ROC curve), whose ordinate is the true-positive rate and abscissa is the false-positive rate. The higher these five metrics are, the better.

4.2 | Experimental results

First, we classified the three original COVID-19 CT datasets using different classifier networks, including the COVID-CT-DenseNet that we designed. In this process, we only used traditional data augmentation methods, such as zooming with a factor of 0.3 and horizontal flipping. Transfer learning was also used in the experiment, as He et al.¹⁰ found that the networks pretrained on large-scale datasets such as ImageNet can achieve better classification performance on COVID-19 CT dataset. Table 3 shows the results of the five metrics we mentioned above.

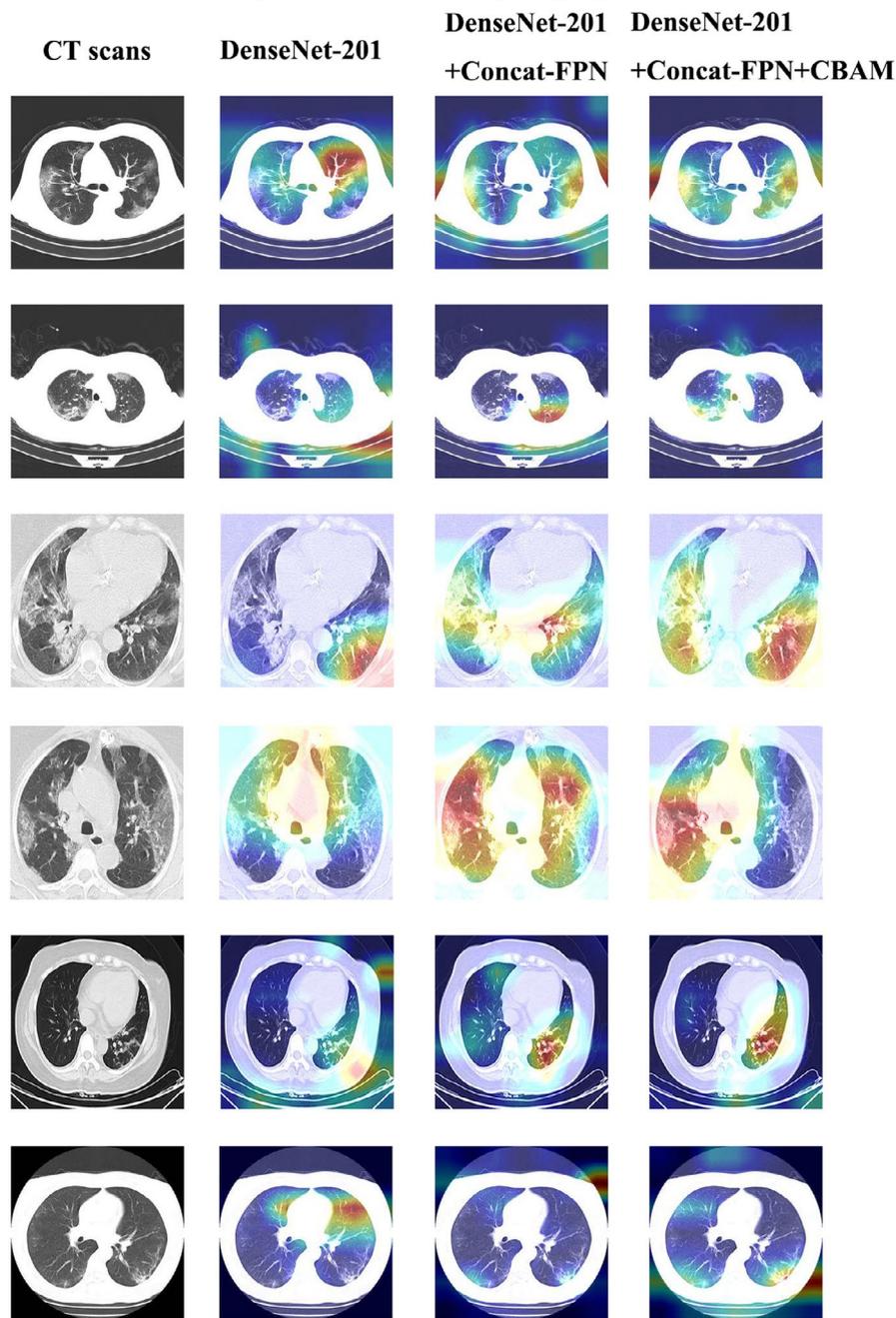


FIGURE 9 The Grad-CAM visualizations for the ablation experiments of COVID-CT-DenseNet. The first and second rows are samples from ①. The third and fourth rows are samples from ②. The fifth and sixth rows are samples from ③ [Color figure can be viewed at wileyonlinelibrary.com]

Figure 7a,b shows the ROC curves of different classifier networks on ② and ③, respectively. The closer the curve is to the upper-left corner, the better the effect. Among the three datasets, ① contains three classes, so we only calculated the AUC of ② and ③. From the comprehensive results, it can be seen that our proposed COVID-CT-DenseNet achieved state-of-the-art classification performance on the three different datasets, compared with other CNN-based classifier networks.

We further used GAN to augment the training data, instead of using traditional methods. In the COVID-CT-GAN we designed, the loss functions used were binary cross entropy and sparse categorical cross entropy. The former calculated the loss

of the generated pseudo-images and the original images, and the latter calculated the category loss of the generated pseudo-images. The training process of the generator and the discriminator is an adversarial process. When the loss of the generator and the discriminator no longer changes or tends to be balanced, it means that the training of GAN is completed.²⁶ We used the trained model to generate images for data augmentation in the classifier. As mentioned above, Frid-Adar et al.⁴⁴ argued that the separated GAN produces better results than the unified GAN. Accordingly, we used the ACGAN, DCGAN and our proposed COVID-CT-GAN to conduct comparative experiments on the three datasets. We used the trained GAN for augmentation, and the

COVID-CT-DenseNet for classification. The results are shown in Table 4. None means that only original data were used for classification. It can be seen that for the three datasets we used, the performance of the five metrics of the unified GAN was better than that of the separated GAN. Examples of the three classes of CT images generated by COVID-CT-GAN are shown in Figure 8. It illustrates that COVID-CT-GAN can learn the characteristics of the three classes of CT images, and the generated images can be used for training the classification network.

5 | DISCUSSIONS

Since there were limited training data for diagnosing COVID-19 on CT images automatically by CNN, an improved architecture of GAN was used to generate images as data augmentation. As is shown in Table 4, it could be found that giving category labels to the generator is beneficial to the quality of generated COVID-19 CT images, and finally improves the classification. Furthermore, when there are more original data in the dataset, the gain achieved by using GAN for data augmentation becomes less obvious, and may have a negative effect, which is

consistent with the conclusion obtained by some of the studies⁴⁵⁻⁴⁸ mentioned above. To further support our proposed method, we conducted ablation experiments on COVID-CT-DenseNet and COVID-CT-GAN. Figure 9 shows the Grad-CAM⁶⁴ visualizations for the ablation experiments of COVID-CT-DenseNet. The experimental results of the metrics are shown in Tables 5 and 6. In addition, we show the number of weight parameters in different networks, as shown in Table 7. In general, the Concat-FPN with an attention mechanism, which we proposed, improved the performance of the original model. As we had previously shown, using Concat-FPN to fuse feature maps of multiple scales directly was slightly coarse, since it could “drown-out” part of the feature information. Therefore, we used the attention mechanism (i.e., CBAM⁶²) to highlight the fused feature information, and the experimental results validated our assumption and analysis. In addition, we also note that the performance gains from applying our proposed Concat-FPN to DenseNet-201 were greater than those from ACGAN. Even the performance of Concat-FPN with the ACGAN was somewhat reduced. We speculate that this may be due to differences between the network architectures. In COVID-CT-DenseNet, the scale of the feature maps was unified through the

TABLE 5 Ablation experiments of COVID-CT-DenseNet

Metrics	Networks	Dataset		
		①	②	③
Accuracy	DenseNet-201	0.82	0.90	0.79
	DenseNet-201 + Concat-FPN	0.85	0.90	0.83
	DenseNet-201 + Concat-FPN +CBAM	0.85	0.91	0.82
Recall	DenseNet-201	0.76	0.83	0.79
	DenseNet-201 + Concat-FPN	0.80	0.91	0.79
	DenseNet-201 + Concat-FPN +CBAM	0.80	0.92	0.85
Precision	DenseNet-201	0.87	0.98	0.79
	DenseNet-201 + Concat-FPN	0.90	0.89	0.85
	DenseNet-201 + Concat-FPN +CBAM	0.88	0.91	0.80
F1-score	DenseNet-201	0.81	0.90	0.79
	DenseNet-201 + Concat-FPN	0.85	0.90	0.81
	DenseNet-201 + Concat-FPN +CBAM	0.84	0.91	0.82
AUC	DenseNet-201	—	0.97	0.87
	DenseNet-201 + Concat-FPN	—	0.97	0.89
	DenseNet-201 + Concat-FPN +CBAM	—	0.97	0.90

TABLE 6 Ablation experiments of COVID-CT-GAN

Metrics	Networks	Dataset		
		①	②	③
Accuracy	ACGAN	0.84	0.92	0.84
	ACGAN +Concat-FPN	0.82	0.89	0.80
	ACGAN +Concat-FPN +CBAM	0.84	0.93	0.85
Recall	ACGAN	0.81	0.94	0.85
	ACGAN +Concat-FPN	0.77	0.92	0.82
	ACGAN +Concat-FPN +CBAM	0.79	0.96	0.87
Precision	ACGAN	0.86	0.93	0.81
	ACGAN +Concat-FPN	0.84	0.92	0.80
	ACGAN +Concat-FPN +CBAM	0.89	0.93	0.83
F1-score	ACGAN	0.83	0.93	0.83
	ACGAN +Concat-FPN	0.80	0.92	0.81
	ACGAN +Concat-FPN +CBAM	0.84	0.94	0.85
AUC	ACGAN	-	0.96	0.92
	ACGAN +Concat-FPN	-	0.92	0.93
	ACGAN +Concat-FPN +CBAM	-	0.98	0.94

convolution with different strides, whereas it is unified through the upsampling with different pooling sizes in the generator of COVID-CT-GAN. In the process of upsampling, the feature loss may be more serious than the process of convolution. In COVID-CT-GAN, we tried using the 3×3 convolution on each merged map to reduce the aliasing effect caused by upsampling as in COVID-CT-DenseNet. However, that greatly increased the number of parameters in the generator, making it difficult to train. This is also the reason why we used Concat-FPN to fuse the

attention maps of different upsampling blocks. The CBAM parameter was very small compared with the generator network, and the shape of its output attention maps was consistent with the input. In the last part of the experiment, we explored a problem that we had encountered, that is, the impact of the number of upsampling layers of the generator and the size of the convolutional kernels on our proposed COVID-CT-GAN. To be consistent with the previous section, we used the generated images for data augmentation, and COVID-CT-DenseNet for classification. For convenience, we denote the five metrics (accuracy, recall, precision, F1-score and AUC) as M1 through M5, respectively. As is shown in Table 8, k is the size of the convolutional kernels and up is the number of upsampling layers of the generator. From the results of the comparison, we found that, as for ① and ②, when the number of upsampling layers of the generator was 3, and the size of the convolution kernels used was 5 or 7, our proposed COVID-CT-GAN could achieve better performance for data augmentation. As for ③, using 7 as the size of the convolution kernels may be better. We think that, in the case of similar network parameters, multiple uses of upsampling may lead to partial feature information loss, whereas less upsampling means fewer initial input feature dimensions, thus affecting the quality of the generated

TABLE 7 Comparison of parameter numbers

Networks	Parameter numbers
VGG-19	20,027,971
EfficientNet-B5	28,527,859
ResNet-101	42,672,515
ResNeXt-101	42,280,899
DenseNet-169	12,654,531
DenseNet-201	18,355,427
COVID-CT-DenseNet	26,972,133
ACGAN	50,768,077
COVID-CT-GAN	53,159,061

TABLE 8 The impact of the number of upsampling layers of the generator and the size of the convolutional kernels on the performance of COVID-CT-GAN

		$k = 3$			$k = 5$			$k = 7$			$k = 9$		
		①	②	③	①	②	③	①	②	③	②	②	③
$up=1$	M1	0.74	0.86	0.77	0.81	0.87	0.79	0.79	0.88	0.80	0.83	0.86	0.77
	M2	0.71	0.87	0.79	0.72	0.89	0.78	0.74	0.90	0.79	0.72	0.89	0.80
	M3	0.81	0.91	0.80	0.83	0.91	0.80	0.85	0.92	0.82	0.86	0.93	0.82
	M4	0.76	0.89	0.79	0.77	0.90	0.79	0.79	0.91	0.80	0.78	0.91	0.81
	M5	—	0.97	0.90	—	0.92	0.89	—	0.90	0.93	—	0.93	0.91
$up=2$	M1	0.80	0.87	0.78	0.82	0.89	0.82	0.81	0.92	0.83	0.82	0.84	0.76
	M2	0.77	0.89	0.82	0.74	0.90	0.86	0.76	0.92	0.88	0.78	0.90	0.81
	M3	0.86	0.90	0.80	0.88	0.94	0.81	0.89	0.94	0.81	0.85	0.91	0.82
	M4	0.81	0.89	0.81	0.80	0.92	0.83	0.82	0.93	0.84	0.81	0.90	0.81
	M5	—	0.94	0.90	—	0.95	0.92	—	0.97	0.94	—	0.96	0.95
$up=3$	M1	0.82	0.88	0.82	0.83	0.90	0.84	0.84	0.93	0.85	0.82	0.91	0.80
	M2	0.73	0.94	0.84	0.80	0.97	0.86	0.79	0.96	0.87	0.77	0.97	0.84
	M3	0.84	0.91	0.79	0.89	0.94	0.83	0.89	0.93	0.83	0.88	0.91	0.84
	M4	0.78	0.92	0.81	0.84	0.95	0.84	0.84	0.94	0.85	0.82	0.94	0.84
	M5	—	0.95	0.91	—	0.97	0.93	—	0.98	0.94	—	0.97	0.93
$up=4$	M1	0.79	0.91	0.80	0.84	0.90	0.83	0.82	0.92	0.84	0.81	0.90	0.79
	M2	0.70	0.93	0.79	0.77	0.92	0.85	0.78	0.95	0.86	0.74	0.93	0.82
	M3	0.83	0.89	0.81	0.84	0.90	0.81	0.90	0.92	0.84	0.87	0.90	0.80
	M4	0.76	0.91	0.80	0.80	0.91	0.83	0.84	0.93	0.85	0.80	0.91	0.81
	M5	—	0.93	0.89	—	0.96	0.93	—	0.96	0.93	—	0.95	0.90

images. It is worth mentioning that in our previous experiments, we used 7 as the size of the convolution kernels, and 3 as the number of upsampling layers in the generator of COVID-CT-GAN.

6 | CONCLUSIONS

In the study, we propose a Concat-FPN architecture with an attention mechanism, which can fuse features of different scales in the network. The Concat-FPN was further used to form COVID-CT-GAN and COVID-CT-DenseNet for generating and classifying COVID-19 CT scan images, respectively. On three COVID-19 CT datasets of different magnitudes, our method achieved improved results in five metrics. We conducted comparative experiments to explore the impact of the number of upsampling layers of the generator and the size of the convolutional kernels on the performance of COVID-CT-GAN. The results demonstrate that by using the new COVID-CT-GAN we could solve the problem of limited training data when using deep learning methods to automatically diagnose COVID-19. Our work improves the accuracy of diagnosing COVID-19 on CT images.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 62063034.

CONFLICT OF INTEREST

The authors have no conflicts to disclose.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in https://pan.baidu.com/s/1TcoP OQ_5TG2gZmxsXOhMkA.

REFERENCES

1. World Health Organization. Coronavirus disease (COVID-19) Weekly Epidemiological and Operational updates November 2020. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports/>
2. Ai T. Correlation of chest CT and RT-PCR testing in coronavirus disease (COVID-19) in China: a report of 1014 cases. *Radiology*. 2019;2020:200642.
3. Fang Y, Zhang H, Xie J, et al. Sensitivity of chest CT for COVID-19: comparison to RT-PCR. *Radiology*. 2020;200432.
4. Shi F, Wang J, Shi J, et al. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19. *IEEE Rev Biomed Eng*. 2020;14:4-15. <https://doi.org/10.1109/RBME.2020.2987975>.
5. Kanne JP, Little BP, Chung JH, Elicker BM, Ketani LH. Essentials for radiologists on COVID-19: an update — radiology scientific expert panel. *Radiology*. 2020;200527.
6. Rodrigues J, Hare SS, Edey A, et al. An update on COVID-19 for the radiologist - a British society of thoracic imaging statement. *Clin Radiol*. 2020;75(5):323-325. <https://doi.org/10.1016/j.crad.2020.03.003>.
7. LeCun Y, Bengio Y, Hinton G, et al. Deep learning. *Nature*. 2015;521(7553):436-444.

8. Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. *Med Image Anal*. 1995;2017(42):60-88.
9. Soares E. SARS-CoV-2 CT-scan dataset: a large dataset of real patients CT scans for SARS-CoV-2 identification. MedRxiv. 2020. <https://doi.org/10.1101/2020.04.24.20078584>.
10. He X, Yang X, Zhang S, et al. Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. MedRxiv. 2020. <https://doi.org/10.1101/2020.04.13.2006394>.
11. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017:2261-2269. <https://doi.org/10.1109/CVPR.2017.243>.
12. He X, Wang S, Shi S., et al. Benchmarking deep learning models and automated model design for COVID-19 detection with chest CT scans. MedRxiv. 2020. <https://doi.org/10.1101/2020.06.08.20125963>.
13. Zhang K, Liu X, Shen J, et al. Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography. *Cell*. 2020;181(6):1423-1433.
14. Hara K, Kataoka H, Satoh Y., et al. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018;6546-6555. <https://doi.org/10.1109/CVPR.2018.00685>.
15. Ouyang XI, Huo J, Xia L, et al. Dual-sampling attention network for diagnosis of COVID-19 from community acquired pneumonia. *IEEE Trans Med Imaging*. 2020;39(8):2595-2605. <https://doi.org/10.1109/TMI.2020.2995508>.
16. Wang S, Kang B, Ma J, et al. A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). MedRxiv. 2020. <https://doi.org/10.1101/2020.02.14.20023028>.
17. Xu X, Jiang X, Ma C, et al. Deep learning system to screen coronavirus disease 2019 pneumonia. 2020; arXiv:2002.09334. [Online]. Available: <http://arxiv.org/abs/2002.09334>.
18. Matsuyama E. A deep learning interpretable model for novel coronavirus disease (COVID-19) screening with chest CT images. *J Biomed Sci Eng*. 2020;13(7):140-152.
19. Song Y, Zheng S, Li L, et al. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. MedRxiv. 2020. Available: <https://doi.org/10.1101/2020.02.23.20026930>.
20. Kang H, Xia L, Yan F, et al. Diagnosis of coronavirus disease 2019 (COVID-19) with structured latent multi-view representation learning. *IEEE Trans Med Imaging*. 2020;39(8):2606-2614. <https://doi.org/10.1109/TMI.2020.2992546>.
21. Anwar T, Zakir S. Deep learning based diagnosis of COVID-19 using chest CT-scan images. TechRxiv. 2020. <https://doi.org/10.36227/techrxiv.12328061>.
22. Kang C, Yu X, Wang S-H, et al. A heuristic neural network structure relying on fuzzy logic for images scoring. IEEE transactions on fuzzy systems; 2020. <https://doi.org/10.1109/TFUZZ.2020.2966163>.
23. Tajbakhsh N, Shin JY, Gurudu SR, et al. Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE Trans Med Imaging*. 2016;35(5):1299-1312. <https://doi.org/10.1109/tmi.2016.2535302>.
24. Greenspan H, van Ginneken B, Summers RM, et al. Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. *IEEE Trans Med Imaging*. 2016;35(5):1153-1159. <https://doi.org/10.1109/TMI.2016.2553401>.
25. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Conference and Workshop on Neural Information Processing Systems (NIPS). 2012;25(2):1097-1105.

26. Goodfellow I. Generative adversarial nets. Conference and Workshop on Neural Information Processing Systems (NIPS); 2014:2672-2680.
27. Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the 34th International Conference on Machine Learning. JMLR. 2016;70:2642-2651.
28. Yi X, Walia E, Babyn P, et al. Generative adversarial network in medical imaging: a review. *Med Image Anal.* 2019;58: 101552.
29. Yi X, Babyn P. Sharpness-aware low dose CT denoising using conditional generative adversarial network. *J Digit Imaging.* 2017;31(5):655-669. <https://doi.org/10.1007/s10278-018-0056-0>.
30. Nie D, Trullo R, Lian J, et al. Medical image synthesis with context-aware generative adversarial networks. In: International Conference on Medical Image Computing and Computer-Assisted InterventionSpringer; 2017:417-425.
31. Jin C-B, Kim H, Liu M, et al. Deep CT to MR synthesis using paired and unpaired data. *Sensors.* 2019;19(10):2361; <https://doi.org/10.3390/s19102361>.
32. Costa P, Galdran A, Meyer MI, et al. End-to-End adversarial retinal image synthesis. *IEEE Trans Med Imaging.* 2018;37(3):781-791. <https://doi.org/10.1109/TMI.2017.2759102>.
33. Yi X, Walia E, Babyn P. Unsupervised and semi-supervised learning with categorical generative adversarial networks assisted by Wasserstein distance for dermoscopy image classification. 2018. arXiv:1804.03700. [Online]. Available: <https://arxiv.org/abs/1804.03700>.
34. Schlegl T, Seeböck P, Waldstein SM, et al. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. International Conference on Information Processing in Medical Imaging, Springer; 2017:146-157. https://doi.org/10.1007/978-3-319-59050-9_12.
35. Hu BO, Tang YE, Chang E-C, et al. Unsupervised learning for cell-level visual representation in histopathology images with generative adversarial networks. *IEEE J Biomed Health Inf.* 2019;23(3):1316-1328. <https://doi.org/10.1109/JBHI.2018.2852639>.
36. Chen X, Duan Y, Houthoofd R., et al. Infogan: interpretable representation learning by information maximizing generative adversarial nets. Conference and Workshop on Neural Information Processing Systems (NIPS); 2016:2172-2180.
37. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of Wasserstein GANs. Conference and Workshop on Neural Information Processing Systems (NIPS); 2017:5767-5777.
38. Springenberg JT. Unsupervised and semi-supervised learning with categorical generative adversarial networks. 2015; arXiv:1511.06390. [Online]. Available: <https://arxiv.org/abs/1511.06390>.
39. Lahiri A, Ayush K, Kumar Biswas P, Mitra P. Generative adversarial learning for reducing manual annotation in semantic segmentation on large scale microscopy images: automated vessel segmentation in retinal fundus image as test case. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2017;794-800. [Online]. Available: <https://www.computer.org/csdl/proceedings-article/cvprw/2017/0733a794/12OmNwvDQts>.
40. Madani A, Moradi M, Karargyris A, Syeda-Mahmood T. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. *Biomed Imaging (ISBI).* 2018;2018:1038-1042.
41. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015, arXiv:1511.06434. [Online]. Available: <https://arxiv.org/abs/1511.06434>.
42. Madani A, Moradi M, Karargyris A, Syeda-Mahmood T. Chest x-ray generation and data augmentation for cardiovascular abnormality classification. Medical Imaging, 2018: Image Processing, 10574. International Society for Optics and Photonics. 2018;2018:57. <https://doi.org/10.1117/12.2293971>.
43. Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. 2018, arXiv:1809.11096. [Online]. Available: <https://arxiv.org/abs/1809.11096>.
44. Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing.* 2018;321:321-331. Available: <https://arxiv.org/abs/1803.01229>.
45. Antoniou A, Storkey A, Edwards H. Data augmentation generative adversarial networks. 2017; arXiv:1711.04340. [Online]. Available: <https://arxiv.org/abs/1711.04340>.
46. Santurkar S, Schmidt L, Madry A. A classification-based study of covariate shifts in GAN distributions. 2017; arXiv:1711.00970. [Online]. Available: <https://arxiv.org/abs/1711.00970>.
47. Li J, Madry A, Peebles J, Schmidt L. On the limitations of first order approximation in GAN dynamics. 2017; arXiv:1706.09884. [Online]. Available: <https://arxiv.org/abs/1706.09884>.
48. Finlayson SG, Lee H, Kohane IS, Oakden-Rayner L. Towards generative adversarial networks as a new paradigm for radiology education. 2018; arXiv:1812.01547. [Online]. Available: <https://arxiv.org/abs/1812.01547>.
49. Karadağ ÖÖ, Erdaş ÇÖ. Experimental assessment of the performance of data augmentation with generative adversarial networks in the image classification problem. Innovations in Intelligent Systems and Applications Conference (ASYU). Izmir. Turkey. 2019;2019:1-4. <https://doi.org/10.1109/ASYU48272.2019.8946442>.
50. Bell S, Zitnick CL, Bala K, Girshick R. Inside-Outside Net: detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016:2874-2883. <https://doi.org/10.1109/CVPR.2016.314>.
51. Kong T, Yao A, Chen Y, Sun F. HyperNet: towards accurate region proposal generation and joint object detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016;845-853. <https://doi.org/10.1109/CVPR.2016.98>.
52. Liu W, Angelov D, Erhan D, et al. SSD: Single shot multi-box detector. In Proceedings of European Conference on Computer Vision. 2016;21-37. https://doi.org/10.1007/978-3-319-46448-0_2.
53. Cai Z, Fan Q, Feris RS, Vasconcelos N. A unified multi-scale deep convolutional neural network for fast object detection. Proceedings of European Conference on Computer Vision. 2016. https://doi.org/10.1007/978-3-319-46493-0_22.
54. Lin T-Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017;1:4. <https://doi.org/10.1109/CVPR.2017.106>.
55. Wang Y, Yu B, Wang L, et al. 3D conditional generative adversarial networks for high-quality pet image estimation at low dose. *NeuroImage.* 2018;174(1):550-562.
56. Larochelle H, Hinton GE. Learning to combine foveal glimpses with a third-order Boltzmann machine. Conference and Workshop on Neural Information Processing Systems (NIPS). 2010;1:1243-1251.
57. Corbetta M, Shulman GL. Control of goal-directed and stimulus-driven attention in the brain. *Nat Rev Neurosci.* 2002;3(3):201-215.
58. Itti L, Koch C, Niebur E, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell.* 1998;20(11):1254-1259. <https://doi.org/10.1109/34.730558>.
59. Rensink RA. The dynamic representation of scenes. *Visual cognition.* 2000;7(1-3):17-42.

60. Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-excitation networks. *IEEE Trans Patt Anal Mach Int.* 2020;42(8):2011-2023. <https://doi.org/10.1109/TPAMI.2019.2913372>.
61. Wang F, Jiang M, Qian C, et al. Residual attention network for image classification. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017;1:6450-6458. <https://doi.org/10.1109/CVPR.2017.683>.
62. Woo S, Park J, Lee JY, Kweon IS. CBAM: Convolutional Block Attention Module. In Proceedings of European Conference on Computer Vision. 2018;11211. https://doi.org/10.1007/978-3-030-01234-2_1.
63. Chollet F. Keras: Deep Learning for Humans. [Online]. Available: 2015. <https://github.com/fchollet/keras>.
64. Selvaraju RR, Cogswell M, Das A, et al. Grad-cam: visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision. 2017;1:618-626.

How to cite this article: Li Z, Zhang J, Li B, Gu X, Luo X. COVID-19 diagnosis on CT scan images using a generative adversarial network and concatenated feature pyramid network with an attention mechanism. *Med. Phys.* 2021;48: 4334–4349. <https://doi.org/10.1002/mp.15044>