




DGCNN: deep convolutional generative adversarial network based convolutional neural network for diagnosis of COVID-19

Saloni Laddha¹ · Vijay Kumar¹ 

Received: 11 June 2021 / Revised: 15 November 2021 / Accepted: 9 February 2022 /
Published online: 8 April 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

The latest threat to global health is the coronavirus disease 2019 (COVID-19) pandemic. To prevent COVID-19, recognizing and isolating the infected patients is an essential step. The primary diagnosis method is Reverse Transcription Polymerase Chain Reaction (RT-PCR) test. However, the sensitivity of this test is not satisfactory to successfully control the COVID-19 outbreak. Although there exist many datasets of chest X-rays (CXR) images, but few COVID-19 CXRs are presently accessible owing to privacy of patients. Thus, many researchers have utilized data augmentation techniques to augment the datasets. But, it may cause over-fitting issues, as the existing data augmentation techniques include small modifications to CXRs. Therefore, in this paper, an efficient deep convolutional generative adversarial network and convolutional neural network (DGCNN) is designed to diagnose COVID-19 suspected subjects. Deep convolutional generative adversarial network (DGAN) consists of two networks trained adversarially such that one generates fake images and the other differentiates between them. Thereafter, convolutional neural network (CNN) is utilized for classification purpose. Extensive experiments are conducted to evaluate the performance of the proposed DGCNN. Performance analysis demonstrates that DGCNN can highly improves the diagnosis performance.

Keywords Chest X-ray · Generative adversarial network · COVID-19 · Convolutional neural network · Data augmentation

1 Introduction

COVID-19 pandemic has taken millions of lives and created havoc worldwide over a short period. COVID-19 is a highly contagious respiratory disease that affects the lungs very badly

✉ Vijay Kumar
vijaykumarchahar@gmail.com

¹ Computer Science and Engineering, National Institute of Technology, Hamirpur, Himachal Pradesh, India

[13]. The only viable solution is to detect the infected individual and break the chain of infection further. Presently, Reverse Transcription Polymerase Chain Reaction (RT-PCR) is widely used for detection of COVID-19 [25]. The expert lab technicians and testing kits are required to perform the testing [3]. The time and expenses involved in testing a sample range from 2 h to a few days. Moreover, RT-PCR does not produce accurate results for some cases due to its high false-negative rates (39–60%) [15]. New variants of SARS-Cov-2 virus have made it even more difficult to detect it by using the existing diagnostic techniques [20].

To overcome the above-mentioned issues, researchers have used artificial intelligence (AI) techniques to develop an automated diagnosis system for coronavirus infection [9]. AI coupled with medical imaging has assisted many fields especially, healthcare in various diagnoses and treatments within the past decade. Recently, deep neural networks [26] are widely used in healthcare system. The well-known deep neural network is convolutional neural network (CNN) [8]. CNNs require huge training data for detecting the coronavirus infection in Chest X-ray (CXR) images [16]. However, the appropriate datasets of CXRs that consists of equal number of COVID-19 and normal chest images is not available [23]. Lack of supervised data may lead to the class imbalance problem [27].

To handle the class imbalance problem, semi-supervised learning techniques are used to perform the classification task [4]. Data augmentation is used to overcome the small-sized dataset problem [24]. The data augmentation includes various operations such as flipping, rotation, and translation. These operations are widely used in most of the computer vision's applications [1]. A new and more efficient form of augmentation is synthetic data augmentation of good-quality samples [2]. Synthetically obtained data samples enrich the dataset and improved the training efficiency. However, it is not possible to generate the completely different images by using the data augmentation [14]. Nowadays, Generative Adversarial Networks (GAN) are used to generate large dataset by generating synthetic data [5].

Deep convolutional GAN (DGAN) [17], which is special type of GAN, is used in this paper. DGAN is used to generate the synthetic COVID-19 CXR images. A novel deep learning-based model is developed for detecting the coronavirus infection on the CXR images. The key contributions of this paper are as follows:

- i. An efficient deep convolutional generative adversarial network and convolutional neural network (DGCNN) is designed to diagnose COVID-19 suspected subjects.
- ii. Deep convolutional generative adversarial network (DGAN) utilizes two networks trained adversarially such that one generates fake images and the other differentiates between them.
- iii. The convolutional neural network (CNN) is utilized for classification purpose.
- iv. Extensive experiments are conducted to evaluate the performance of the proposed DGCNN.

The remaining structure of this paper is as follows. Section 2 presents the related work done in the direction of COVID-19 diagnosis. The proposed deep learning-based model is described in Section 3. The experimental results and discussion are mentioned in Section 4. The concluding remarks and possible future research directions are presented in Section 5.

2 Related work

Many studies have implemented a supervised learning framework to develop CNN models for COVID-19 diagnosis using CXRs. Hemdan et al. [13] integrated seven convolution networks

and proposed a COVIDX-Net model in CXRs. COVIDX-Net provided better performance than the individual models. However, the efficiency of COVIDX-Net was tested on very small dataset. Wong [25] developed a COVIDNet to detect and classify pneumonia cases along with COVID-19. This model attained the classification accuracy of 92.4%. Ioannis [3] expanded the dataset taking 224 COVID-19 positive cases. This model obtained a 98.75% success rate for binary and 93.48% accuracy for triple-classes. Narin [15] coupled ResNet-50 with CXRs and achieved a 98% accuracy in diagnosis. Sethy and Behra [20] used feature classification of images obtained from various deep networks with a support vector machine (SVM) classifier. Besides these, many recent studies are using deep learning that worked on Computed Tomography (CT) of lungs to classify and detect COVID-19 cases. However, the lack of publicly available datasets was an issue in all the deep learning models as they were trained on very small datasets.

A few more experiments were conducted by DeGrave [9] to test the robustness and generalizability of CNN models. They achieved good test accuracy on replicating various supervised models like COVID-Net [26] on the COVID-x dataset. However, their predictive performance falls by 50% on validating the model on external COVID and non-COVID datasets [8]. They used saliency maps and image edges for detecting COVID-19 [16]. It is found that non-COVID markers did not provide the prediction such as cardiac silhouette and diaphragm. The images obtained from different scanners can be problematic as the models with high accuracy on a particular dataset cannot be generalized for other datasets.

The above-mentioned data inadequacy in medical diagnosis led to explore new ways to expand image datasets. Recently, GANs are utilized by many researchers in medical imaging [23]. Zhao [27] proposed a VGG-16 network with DGAN for synthetic image generation in lung-nodule classification. A progressively grown GAN (PGGAN) was trained to synthesize medical scans of the fundus containing premature retinopathy vascular pathology (ROP) and MRI scans [4]. Waheed et al. [24] presented an auxiliary classifier GAN (ACGAN)-based model named as CovidGAN to generate the synthetic CXR images for COVID-19 classification. CovidGAN improved the performance of CNN. However, the performance can be further by using PGGAN. Acar et al. [1] hybridized GAN, data augmentation and segmentation to improve the performance of COVID-19 classifier. The synthetic CT images generated from hybrid approach were applied in CNN for identification of COVID-19. The performance of classifier was little improvement by utilizing the proposed hybrid approach. Al-Shargabi et al. [2] utilized a conditional GAN for generating synthetic CXR images. These images were used for training the InceptionResNet_V2 model and provided better performance than the InceptionResNet_V2 without GAN. However, the model has high computational time. Schlegl [19] studied the data distribution of healthy tissues in the retina using GAN. It was used for anomaly detection in the retinal region on new and healthy image patches.

It is observed from the above-mentioned literature that GAN-based techniques can be useful to detect the infection caused by COVID-19. Studies showed that the anomalies in chest radiographs are more frequent during the initial symptom period and at the peak of illness. Thus, CXR should be used as a major screening method for COVID-19 diagnosis. It has other advantages such as easy accessibility and portability. The less number of CXR images are available due to small duration of this pandemic. Hence, GAN can be used to create the synthetic training data. CXR images can be synthesized from scratch using GAN and produce the desirable results when it is combined with other methods. These fact motivated us to develop the novel deep learning-based model for identification of COVID-9 in CXR images.

3 Materials and methods

3.1 Dataset

CXR's dataset is obtained from [7] for experimental purpose. It is a well-structured dataset that consists of four classes namely, COVID-19, normal, pneumonia viral, and pneumonia bacterial. It has 306 images in total that are further divided into 270 training images and 36 test images, respectively (refer [7]). Figure 1 shows a sample view of CXR's testing dataset. Table 1 shows the subject wise distributions of the dataset. It shows that the dataset is balanced in nature. The only problem with this dataset is limited number of images for deep learning models. To produce synthetic images for each of the four classes, CXRs dataset [7] is utilized as an input to DGAN with real images. The obtained images and actual CXRs images are used to train the proposed model.

3.2 Proposed architecture

To augment the dataset, DGAN is used for obtaining the synthetic images for each of the four classes of the original dataset. Further, CNN classifier is used to evaluate the effect of synthetic images on the classification performance.

3.2.1 Generative adversarial network (GAN)

GANs are the specific frameworks of a generative model. A generative model learns the data distribution (p_{data}) to obtain new samples from a given set of sample images x_1, x_2, \dots, x_n . DGAN is used when both discriminator and generator are deep CNNs [11]. It consists of two

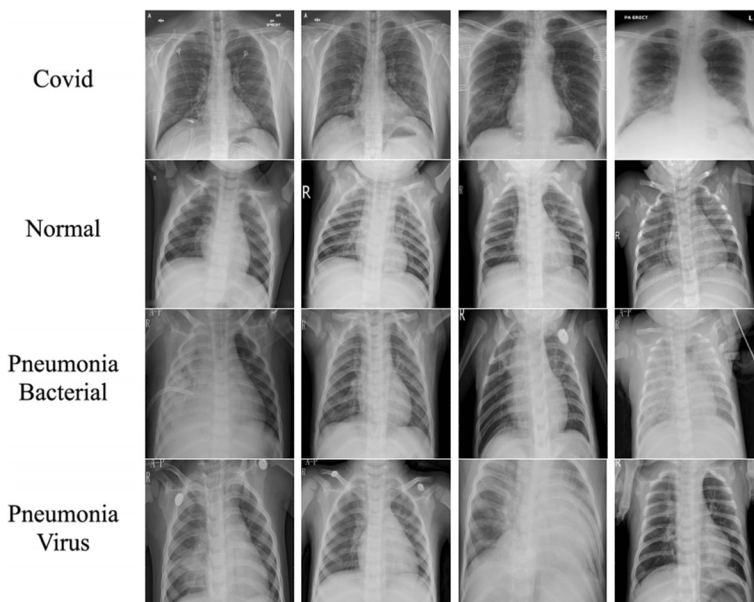


Fig. 1 Sample view of CXR's testing dataset

Table 1 Subjects wise distribution of the dataset

Set/Class	COVID-19	Normal	Bacterial pneumonia	Viral pneumonia	Total
Training	60	70	70	70	270
Testing	9	9	9	9	36
Total	69	79	79	79	306

simultaneously trained networks, i.e., discriminator (D) and the generator (G). D discriminates between the obtained real and fake images. It takes input as i and outputs the probability of the real sample as $D(i)$. The generator network takes input j_1, \dots, j_n from a uniform distribution $p(i)$ and synthesizes samples after mapping $G(j)$ to the image space $p(g)$. G targets to obtain $f(g) = f(data)$. In a two player minimax game, the loss function is optimized to train the adversarial networks as:

$$\min_G \max_D E_{i \sim f(data)} \log D(i) + E_{i \sim f(i)} [\log (1 - D(G(j)))] \quad (1)$$

D is trained such that $D(i)$ is maximized for images having $i \sim f(data)$ and is minimized for images having $i = f(data)$. G outputs images $G(z)$ and deceives the discriminator during the training process such that $D(G(j)) \sim f(data)$. Hence, the generator is trained to maximize $D(G(j))$, or minimize $1 - D(G(j))$ [6]. G boosts its capacity to synthesize more realistic images during training. While D boosts its capacity to differentiate the real from the synthesized images.

a. Generator

An input vector consisting of 100 random numbers is given to the generator from a uniform distribution, which produces the output of $64 * 64 * 1$ sized image. The network architecture is made up of a fully connected layer that is resized to $4 * 4 * 512$ and four fractional-stride convolutions to up-sample the image with $4 * 4$ sized kernels. A fractional stride convolution is expanded by inserting zeros in between the pixels. Besides the output layer, Batch-normalization is applied to every layer. The learning process of DGAN is stabilized by normalizing the responses throughout the mini-batch, which stops the generator from reducing all samples to a single point. *Tanh* activation function is used by the output layer and *ReLU* for all the other layers. Figure 2 shows the generator network architecture along with the trainable parameters.

b. Discriminator

The discriminator network inputs CXR images with size of $64 * 64 * 1$ and outputs one decision if CXR is real. Figure 3 depicts the architecture of discriminator. It consists of four convolutions and fully connected layers with $4 * 4$ sized kernels. The spatial dimensionality is reduced by using stride convolutions. Leaky Rectified Linear Unit (ReLU) function $f(i) = \max(i, leak * i)$ and batch normalization is used for all layers except the input-output. The sigmoid function is used in the output for the likelihood probability score of CXR as $[0, 1]$. Also, to prevent the over-fitting issue, the dropout is used for every layer, except the second and the output layers.

Layer (type:depth-idx)	Output Shape	Param #
Sequential: 1-1	[-1, 1, 64, 64]	--
└─ConvTranspose2d: 2-1	[-1, 512, 4, 4]	819,200
└─BatchNorm2d: 2-2	[-1, 512, 4, 4]	1,024
└─ReLU: 2-3	[-1, 512, 4, 4]	--
└─ConvTranspose2d: 2-4	[-1, 256, 8, 8]	2,097,152
└─BatchNorm2d: 2-5	[-1, 256, 8, 8]	512
└─ReLU: 2-6	[-1, 256, 8, 8]	--
└─ConvTranspose2d: 2-7	[-1, 128, 16, 16]	524,288
└─BatchNorm2d: 2-8	[-1, 128, 16, 16]	256
└─ReLU: 2-9	[-1, 128, 16, 16]	--
└─ConvTranspose2d: 2-10	[-1, 64, 32, 32]	131,072
└─BatchNorm2d: 2-11	[-1, 64, 32, 32]	128
└─ReLU: 2-12	[-1, 64, 32, 32]	--
└─ConvTranspose2d: 2-13	[-1, 1, 64, 64]	1,024
└─Tanh: 2-14	[-1, 1, 64, 64]	--
Total params: 3,574,656		
Trainable params: 3,574,656		
Non-trainable params: 0		
Total mult-adds (M): 423.53		
Input size (MB): 0.00		
Forward/backward pass size (MB): 1.91		
Params size (MB): 13.64		
Estimated Total Size (MB): 15.54		

Fig. 2 Generator architecture in DGAN

3.2.2 Classifier

The architecture of the proposed classifier and the relative trainable parameter are demonstrated in Fig. 4. CNN architectures for medical imaging generally consist of lesser convolution layers due to small datasets and small input sizes. In this paper, CNN classifier utilizes fixed size grayscale input CXRs of $64 * 64$ that is normalized in the range of $[0, 1]$.

The network architecture is a three-layered convolution with $3 * 3$ kernel, ReLU activation functions, batch normalization, and max pooling. The output layer have two fully connected layers with dropout as 0.5 and Softmax output function over the four classes. Batch normalization and dropout are utilized to prevent over-fitting.

3.2.3 Model building

After defining the network architecture, this section explains the training of the proposed DGCNN model. It involves the selection of appropriate loss function, selecting epoch value, and various learning rates. DGANs are capable of producing extremely good results, but training a DGAN is quite challenging as both the sub-networks are trained simultaneously and may affects the training of each other. It is often the case that improvements to one sub-network come at the expense of the other sub-network.

CNN architectures for medical imaging generally consist of lesser number of convolution layers due to lesser size of the datasets and small input sizes [22]. In this paper, the training process is extended until the loss curve stabilizes around its minimum axis. It may depend on the size of the dataset. Since the dataset length differs, therefore, epochs are allowed to be 1500.

Layer (type:depth-idx)	Output Shape	Param #
Sequential: 1-1	[-1, 1, 1, 1]	--
└─Conv2d: 2-1	[-1, 64, 32, 32]	1,024
└─LeakyReLU: 2-2	[-1, 64, 32, 32]	--
└─Dropout: 2-3	[-1, 64, 32, 32]	--
└─Conv2d: 2-4	[-1, 128, 16, 16]	131,072
└─BatchNorm2d: 2-5	[-1, 128, 16, 16]	256
└─LeakyReLU: 2-6	[-1, 128, 16, 16]	--
└─Conv2d: 2-7	[-1, 256, 8, 8]	524,288
└─BatchNorm2d: 2-8	[-1, 256, 8, 8]	512
└─LeakyReLU: 2-9	[-1, 256, 8, 8]	--
└─Dropout: 2-10	[-1, 256, 8, 8]	--
└─Conv2d: 2-11	[-1, 512, 4, 4]	2,097,152
└─BatchNorm2d: 2-12	[-1, 512, 4, 4]	1,024
└─Dropout: 2-13	[-1, 512, 4, 4]	--
└─LeakyReLU: 2-14	[-1, 512, 4, 4]	--
└─Conv2d: 2-15	[-1, 1, 1, 1]	8,192
└─Sigmoid: 2-16	[-1, 1, 1, 1]	--
Total params: 2,763,520		
Trainable params: 2,763,520		
Non-trainable params: 0		
Total mult-adds (M): 104.48		
Input size (MB): 0.02		
Forward/backward pass size (MB): 1.38		
Params size (MB): 10.54		
Estimated Total Size (MB): 11.93		

Fig. 3 Discriminator architecture in DGAN

i. Batch Size

For training DGAN, larger batch size is not recommended. Because in the initial phases of training the discriminator gets a lot of examples to train. Also, a larger batch size can overpower the generator. It may lead to an unstable model and it may not converge properly. Considering the size of the dataset, the batch size is taken as 32. It is quite suitable since both the discriminator as well as the generator could train with stability.

For classifier, the strategy used is operating in a small-batch regime wherein the set of training data, a subset of 256 elements is randomly sampled to obtain an approximation to the gradient. The quality of the model is significantly degraded for a larger batch size. The batch size strategies are evaluated and it seems to be the best suited for the proposed DGCNN.

ii. Transform of training samples

To train CNN, separate sets of training datasets were developed using classical augmented and synthetic augmented CXRs. Various image transforms are utilized together before assigning the training samples to DGCNN:

- **Resize:** It is used to resize the input image to the given size. Since DGAN takes images having size $64 * 64$, thus, the input images are resized $64 * 64$ to before providing them to DGCNN.
- **Random horizontal flip:** It can augments the images by flipping them horizontally with random degree.

Layer (type:depth-idx)	Output Shape	Param #
Sequential: 1-1	[-1, 32, 32, 32]	--
└─Conv2d: 2-1	[-1, 32, 64, 64]	320
└─ReLU: 2-2	[-1, 32, 64, 64]	--
└─BatchNorm2d: 2-3	[-1, 32, 64, 64]	64
└─MaxPool2d: 2-4	[-1, 32, 32, 32]	--
Sequential: 1-2	[-1, 64, 16, 16]	--
└─Conv2d: 2-5	[-1, 64, 32, 32]	18,496
└─ReLU: 2-6	[-1, 64, 32, 32]	--
└─BatchNorm2d: 2-7	[-1, 64, 32, 32]	128
└─MaxPool2d: 2-8	[-1, 64, 16, 16]	--
Sequential: 1-3	[-1, 128, 8, 8]	--
└─Conv2d: 2-9	[-1, 128, 16, 16]	73,856
└─ReLU: 2-10	[-1, 128, 16, 16]	--
└─BatchNorm2d: 2-11	[-1, 128, 16, 16]	256
└─MaxPool2d: 2-12	[-1, 128, 8, 8]	--
Sequential: 1-4	[-1, 4]	--
└─Flatten: 2-13	[-1, 8192]	--
└─Linear: 2-14	[-1, 256]	2,097,408
└─ReLU: 2-15	[-1, 256]	--
└─Dropout: 2-16	[-1, 256]	--
└─Linear: 2-17	[-1, 4]	1,028
└─Softmax: 2-18	[-1, 4]	--
Total params: 2,191,556		
Trainable params: 2,191,556		
Non-trainable params: 0		
Total mult-adds (M): 43.22		
Input size (MB): 0.02		
Forward/backward pass size (MB): 3.50		
Params size (MB): 8.36		
Estimated Total Size (MB): 11.88		

Fig. 4 Proposed architecture of CNN classifier

- **Grayscale:** This function converts the input image to a single channel which is later assigned to DGCNN.
- **Normalize:** The given set of CXR images are normalized to have unit mean as well as unit standard deviation.
- **Color Jitter:** It is used to alter saturation, brightness, and contrast of CXR images.
- **Random Rotation:** It is used to randomly rotate CXRs depending upon the predefined interval range of degrees.
- **Random Affine:** It is used for translation and scaling to CXR images in the predefined range.

iii. Hyper parameters

Four separate DGANs [21] are trained to synthesize images for each of four type classes. Both the networks are trained iteratively. Mini-batches are drawn of $n = 32$ CXR samples $x(1), \dots, x(n)$ for each CXR class \in (COVID, Normal, Pneumonia bacteria, and Pneumonia virus) and $n = 32$ noise samples $y(1), \dots, y(n)$ from a uniform distribution. The slope of leak was 0.2 in Leaky ReLU. Adam optimizer with parameters $\alpha_1 = 0.5$ and $\alpha_2 = 0.999$ are used. The initial learning rate was 0.0002 for both networks. Different combinations of learning rates for generator and discriminator networks are used to analyze the convergence of DGAN. Finally, the learning rate of generator is increased to 0.002.

For training the classifier, batch size is used as 256. Stochastic gradient descent optimization with a high learning rate ($LR = 0.01$) is used that performed better for this specific task. Nesterov momentum updates ($m = 0.9$) and gradient at lookahead position are used, which speed up the training process and improve the performance. Separate CNNs are trained on different augmented CXR scans for each of the available datasets for variable epoch's quantities depending on the dataset dimensions, i.e., 150 to 750 epochs depending upon the size of the dataset.

4 Experimental setup and analysis

This section discusses the performance metrics, comparative analysis of the proposed DGCNN model. For experimental purpose, an open-source machine learning library PyTorch is used. Google Colab is used to train both DGAN and classifier models.

4.1 Evaluation metrics

For evaluating the performance of the proposed DGCNN, the training performance can be evaluated as:

- **Loss_D:** It represents the discriminator loss and is taken as the sum of losses for all the real and fake batches that given as: $(\log(D(G(j))) + \log(D(i)))$.
- **Loss_G:** It is the generator loss and can be taken as $\log(D(G(j)))$.
- **$D(x)$:** It represents the average discriminator output for all real batches. It initializes using 1 and then theoretically can converge to 0.5 according to the value of G .
- **$G(z)$:** It is the average discriminator output for all the fake batches. The first value is $(D - G - j1)$ before updating D and the second value is $(D - G - j2)$ after updation of D . $D - G - j2$ is initialized as 0 and later converges to 0.5 according to the value of G .

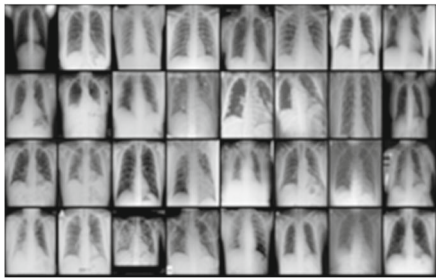
Whenever a discriminator is updated, it tries to push $D(i)$ towards 1 and at a same time pushes $D(G(j))$ towards 0. An updated generator tries to increase $D(G(j))$, i.e., it tries to dupe D that the images generated from noise are the real ones. The discriminator cannot differentiate between real and fake images in an ideal case [12]. However, this scenario is not easily achieved practically.

To evaluate the classifier, the number of CXRs in the training set is (9, 9, 9, and 9) for all kinds of subjects. Therefore, a batch of 36 CXRs is used to maintain the balance of CXR images in every class. The average performance is computed over 500 iterations. The average testing accuracy value is used to evaluate the performance of DGAN. Additional measures such as sensitivity, specificity, and F1-score are also computed for each category.

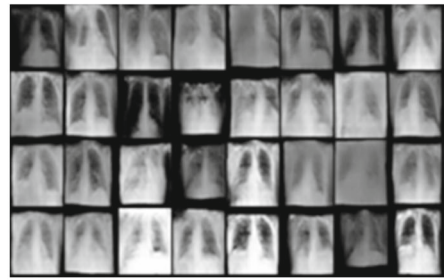
4.2 Performance analysis

4.2.1 Fake images generation using DGAN

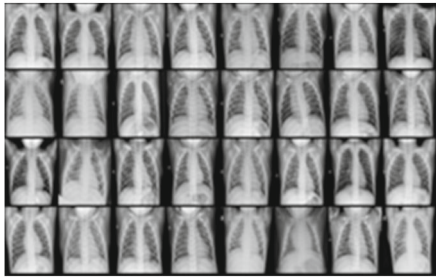
COVID-19 CXR [7] dataset is used for training DGAN. For analysis of every trained network, the relevant parameters namely, $D(G(j1))$, $D(G(j2))$, $D(i)$, $Loss - D$, and $Loss - G$, are obtained and plotted against the epochs. Figure 5 shows the performance of a stack of real



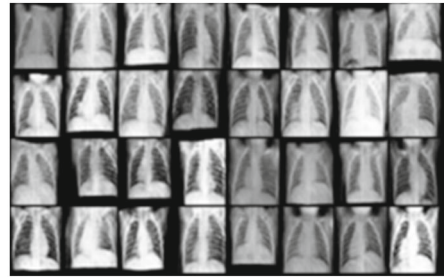
(a) COVID real



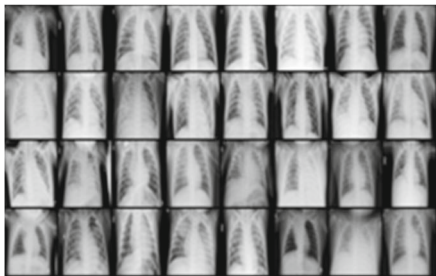
(b) COVID fake



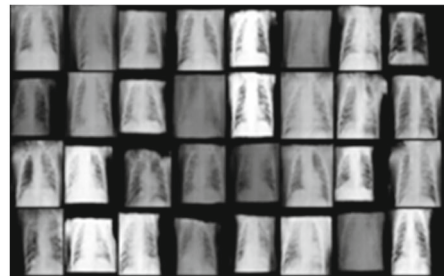
(c) normal real



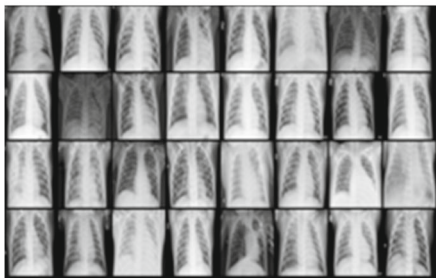
(d) normal fake



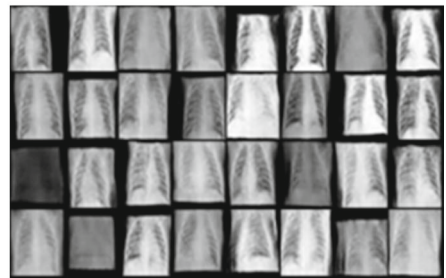
(e) bacterial pneumonia real



(f) bacterial pneumonia fake



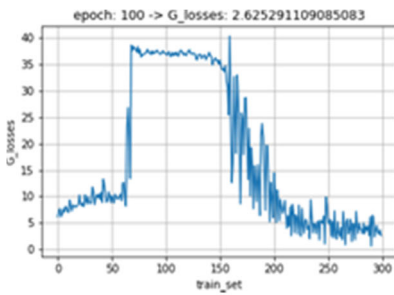
(g) viral pneumonia real



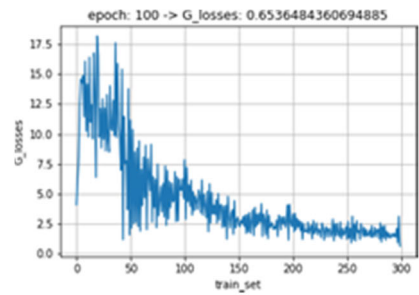
(h) viral pneumonia fake

Fig. 5 Pairs of real (left) and fake (right) images generated using DGAN

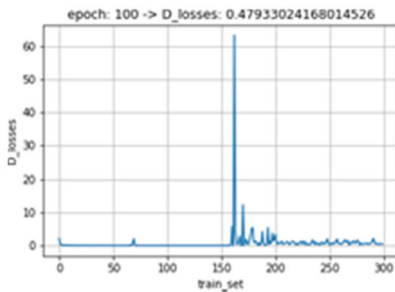
CXRs from the dataset and fake CXRs generated from DGAN trained. It is found that the stability of both networks is increased as the epochs increased. The fake images are shown in



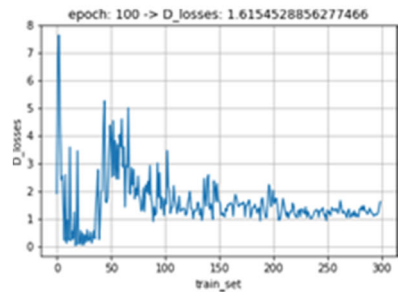
(a) G_loss (LR=0.0002)



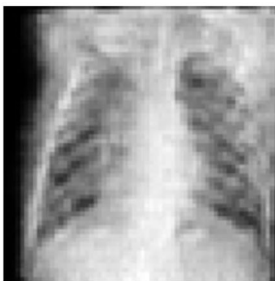
(b) G_loss (LR=0.002)



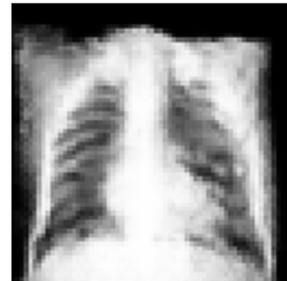
(c) D_loss (LR=0.0002)



(d) D_loss (LR=0.002)



(e) Generated image (LR=0.0002)



(f) Generated image (LR=0.002)

Fig. 6 Analysis of the learning rate: **a** and **b** generator loss analysis with respect to epochs, **c** and **d** discriminator loss analysis, and **e** Generated fake image for the generator network when LR = 0.0002, and **f** generated fake image when LR = 0.002

the right portion of Fig. 5, which are single-channel images obtained from an individual DGAN trained with 512 epochs for each of the four classes.

4.2.2 Comparison for different learning rates

Four DGANs are trained by considering every individual class of the dataset. Initially, a low learning rate of 0.0002 for the generator network resulted in the poor quality of synthetic

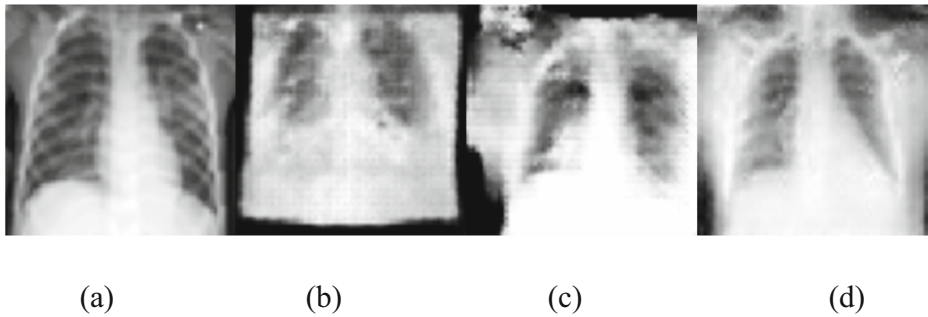


Fig. 7 Epochs analysis **a** Actual pneumonia virus CXR. **b** Generated image at epoch 50. **c** Generated image at epoch 100. **d** Generated image at epoch 200

images. The major cause is the discriminator overpowering the generator network. The output synthetic images from the network are improved in quality when the learning rate is increased to 0.002. Thus, DGAN perform efficiently on the used dataset when the learning rate is 0.002 (see Fig. 6a–d).

4.2.3 Analysis of epochs

The performance of DGAN is also evaluated by using the epoch values as 128, 256, and 512. Figure 7 shows the obtained fake images using the different epoch values. The quality of images is increased as the number of epochs are increased. It is found that DGAN produce fake images, which are closer to the actual CXR images for larger value of epoch.

4.2.4 Convergence failure

During the training process, if the generator and the discriminator fail to reach a balance, it may results in a convergence failure. In the case of discriminator dominates, the generator score approaches to 0 and the discriminator score approaches to 1. It overpowers the generator as shown in Fig. 8.

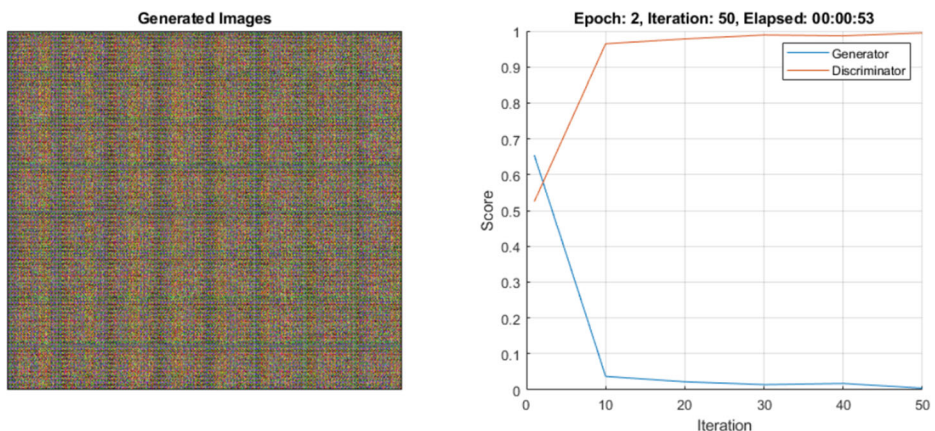


Fig. 8 Discriminator overpowering the generator

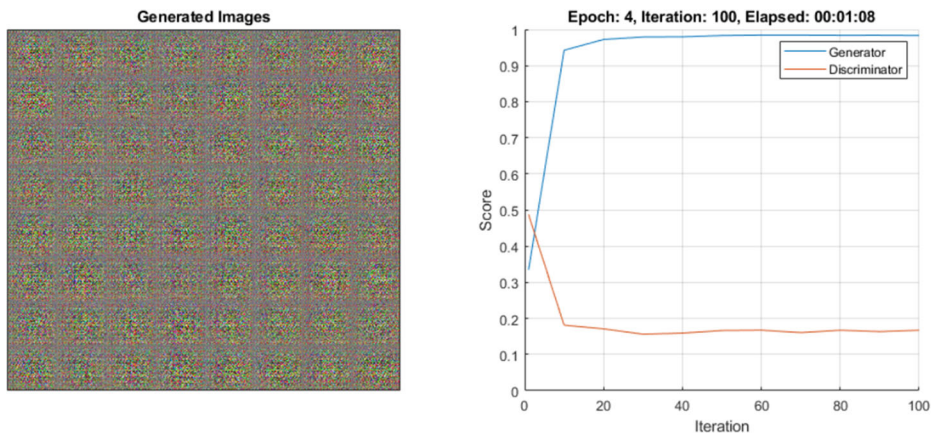


Fig. 9 Generator overpowering the discriminator

In contrast to the discriminator dominates, in the case of generator dominates the generator score approaches to 1. The score remains near to 1 for many iterations and the discriminator is duped by the generator almost every time. Figure 9 shows the case of generator dominates where the generator overpowers the discriminator.

4.2.5 Classification analysis

Figure 10 shows the flowchart of the proposed DGCNN for the evaluation of synthetic data augmentation to diagnosis the suspected cases. Initially, the performance of existing data augmentation techniques is evaluated. DGAN is then used to synthesize CXR scans and the

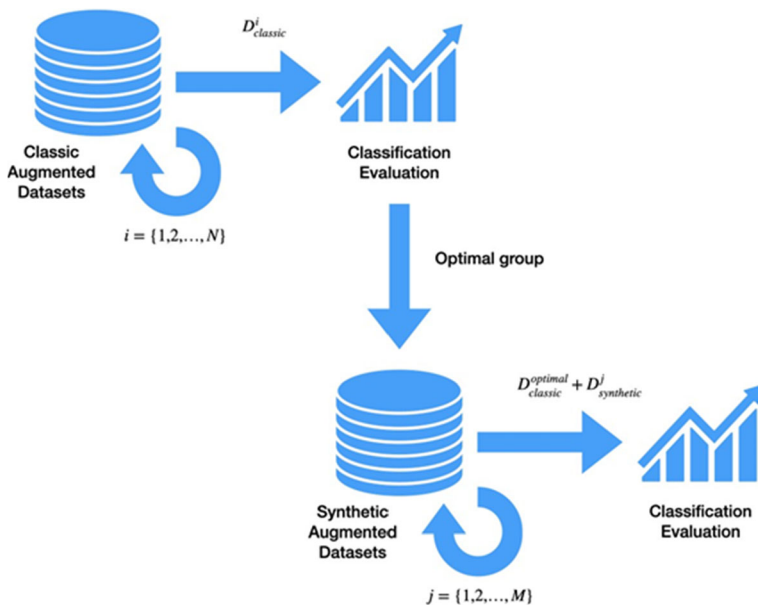


Fig. 10 Flowchart of the proposed DGCNN for synthetic data augmentation to diagnose COVID-19 suspected cases

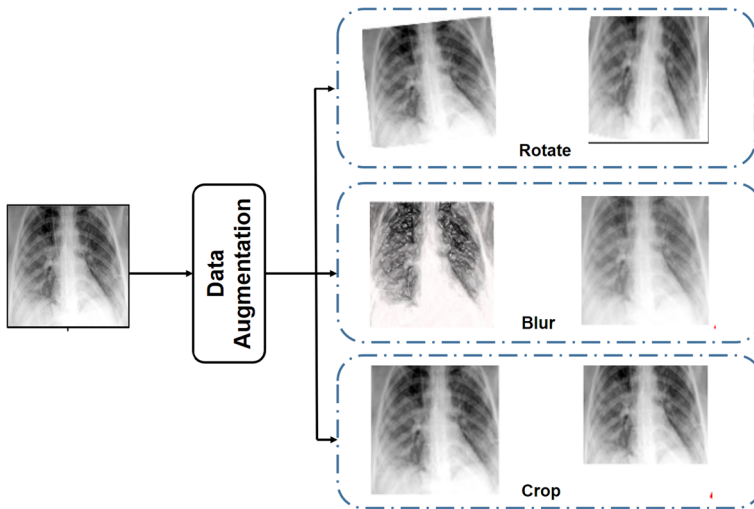


Fig. 11 Sampled images after the data augmentation

resultant images are combined with the actual CXR scans for training purpose. Subsequent section discusses the various steps of the proposed DGCNN for COVID-19 diagnosis.

Step 1: Existing augmentation In this step, the classification results of CNN model are evaluated on actual and data augmentation-based training dataset. CNN is trained and the respective performance is evaluated separately for both sets of data, i.e., actual and augmented CXRs as. Let $D_{classic}$ represents the training data that includes an augmented CXR scans for training. Some fraction of CXR scans is also used for evaluations during the testing time. Additional data groups are formed for examining the effect of increased samples. First data group consist of only actual CXR scans. Various data augmentations are utilized ($N_{rot} = 2$, $N_{color} = 2$, $N_{flip} = 2$, $N_{scale} = 4$, and $N_{trans} = 4$) for each original scan. It results in $N = 128$ augmented images per CXR scan. Therefore, 8000 samples per class are obtained. Thereafter, the images are selected by sampling random augmented scans such that same augmentation volume is sampled for each original scan. To summarize this data group preparation process in augmentation, 500, 1000, 2000, and 3000 samples are added, respectively to each fold. The training process is cross-validated over 4 different folds. Fig. 11 shows the sampled images after the data augmentation.

Step 2: Synthetic augmented datasets In the second step, synthetic CXR scans are generated for data augmentation using DGAN. The optimal point for classic augmentation $D_{classic}^{optimal}$ is

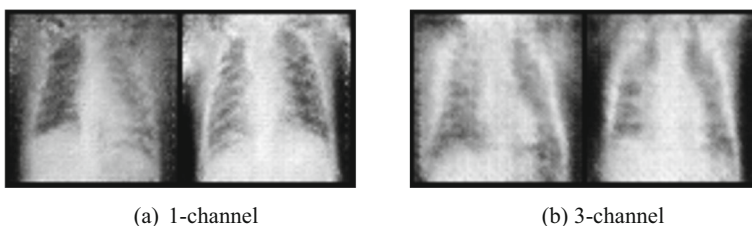


Fig. 12 GAN trained generated images for bacterial pneumonia CXRs

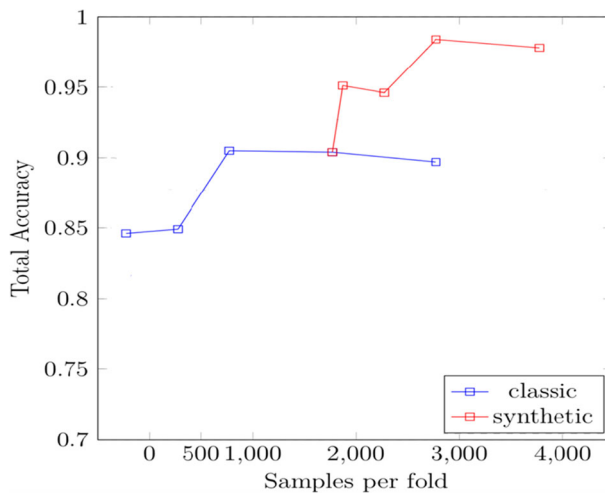


Fig. 13 Accuracy analysis of existing data augmentation and DGAN based CXR scans with respect to increase in training set size

taken and the augmented data group is used for training DGAN. For effective training, the existing data augmentation is incorporated because of the small dataset. DGAN [10, 18] is employed for training each class separately using the same data fraction. The generator synthesized new samples after separate learning of each class data distribution. Some examples of synthesized CXR scans from each class are presented in Fig. 12. The same approach is used for constructing data groups. Additionally, numerous synthetic scans for all four classes are collected and data groups $D_{synthetic}$ of synthetic data are evaluated. The same number of synthetic scans are sampled for every class to keep them balanced. To summarize the data group preparation process in synthetic augmentation, 100, 500, 1000, and 2000 samples are appended to each of the four-folds.

Figure 13 shows the experimental results obtained from DGAN for synthetic augmentation. The baseline results for the existing data augmentation techniques are marked as red. Total accuracy measure for each group of data for CXR diagnosis is evaluated. The average test results of CNN prediction over 500 iterations are reported in tables. The blue line demonstrates the result of the existing data augmentation scenario. The red line shows the result obtained from the combined approach of both synthetic data augmentation and the existing data augmentation base CXR scans. It results in 76.9% accuracy when no augmentation is used. This is happened due to the over-fitting problem. Table 2 shows the performance evaluation of proposed model without using data augmentation over 750 epochs.

Table 2 Performance analysis of CNN model on training set without data augmentation

Model/Metric	Accuracy	Sensitivity	Specificity	F1-score
COVID-19	0.9119	0.9055	0.9202	0.9116
Normal	0.9152	0.9172	0.9068	0.9142
Bacterial Pneumonia	0.9165	0.9192	0.9172	0.9135
Viral Pneumonia	0.9113	0.9149	0.9205	0.9107
Average	0.9137	0.9142	0.916175	0.9125

Table 3 Performance analysis of CNN model on training dataset with the existing data augmentation

Model/Metric	Accuracy	Sensitivity	Specificity	F1-score
COVID-19	0.9493	0.9472	0.9427	0.9442
Normal	0.9379	0.9346	0.9388	0.9358
Bacterial Pneumonia	0.9397	0.9362	0.9404	0.9385
Viral Pneumonia	0.9366	0.9456	0.9454	0.9341
Average	0.9408	0.9409	0.9418	0.9381

Table 3 shows performance analysis of CNN model when it is trained with 750 epochs on training dataset by using the existing data augmentation techniques. It clearly shows that there is an improvement in terms of average accuracy as 1.7463%. The main reason behind this improvement is that the impact of over-fitting is reduced due to the increase in the training dataset.

Table 4 demonstrates performance analysis of CNN model when it is trained on existing data augmentation and DGAN based training dataset with 750 epochs. It clearly shows that there is an improvement in terms of average accuracy as 5.8472% as compared to the without the use of any data augmentation technique and 4.1009% as compared to the existing data augmentation-based CNN model, respectively. The main reason behind this improvement is that the impact of the over-fitting is significantly reduced due to the major improvement in the size of the training dataset.

4.3 Discussion

Table 5 shows the comparative analysis of the proposed and the existing COVID-19 diagnosis models such as CSEN [34], TL-SVM [36], COVID-Net [33], MC-ResNet [40], COVID-DenseNet [39], COVIDX-Net [32], CNN-SVM [31], ResExLBP-SVM [30], COVIDiagnosis-Net [29], Xception and ResNet50V2 [28], ChestNet [18], MobileNet and SqueezeNet-based SVM [5], Xception [12], CovXNet [14], and DCNN [10]. Compared to these models the proposed DGCNN model achieves significantly better accuracy. It is also observed from results that the significant improvement has been achieved by the proposed model by using both existing data augmentation and synthetic data augmentation generated by DGAN. The improvement in average accuracy is 5.8472% as compared to the without the use of any data augmentation technique. Whereas, the improvement in average accuracy is 4.1009% as compared to the existing data augmentation-based CNN model, respectively.

Table 4 Performance analysis of CNN model on training dataset with both existing data augmentation and synthetic augmentation

Model/Metric	Accuracy	Sensitivity	Specificity	F1-score
COVID-19	0.9922	0.9827	0.9866	0.9838
Normal	0.9912	0.9828	0.9921	0.9832
Bacterial Pneumonia	0.9835	0.9819	0.9885	0.9802
Viral Pneumonia	0.9886	0.9884	0.9851	0.9923
Average	0.9889	0.9839	0.9881	0.9849

Table 5 Accuracy analysis among the existing and the proposed DGCNN models

Model	Accuracy
CSEN [34]	95.9%
TL-SVM [36]	94.70%
COVID-Net [33]	93.30%
MC-ResNet [40]	95.50%
COVID-DenseNet [39]	93.71%
COVIDX-Net [32]	90.00%
CNN-SVM [31]	95.20%
ResExLBP-SVM [30]	99.29%
COVIDiagnosis-Net [29]	98.30%
Xception and ResNet50V2 [28]	91.40%
ChestNet [18]	94.67%
MobileNet and SqueezeNet-based SVM [5]	98.25%
Xception [12]	97.41%
CovXNet [14]	97.40%
DCNN [10]	94.48%
Proposed DGCNN	98.89%

5 Conclusion

It has been found that the sensitivity of RT-PCR test is not satisfactory to successfully control the COVID-19 outbreak. Therefore, an efficient DGCNN has been designed to diagnose COVID-19 suspected subjects. In the proposed DGCNN model, initially, DGAN consists two networks that has been trained adversarially such that one generates fake images and the other differentiates between them. Thereafter, CNN has been utilized to diagnose suspected cases from CXR scans. Extensive experiments have been drawn to evaluate the performance of the proposed DGCNN. Performance analysis have shown that DGCNN can highly improves the diagnosis performance. It has been found that the proposed DGCNN achieved an improvement in terms of average accuracy as 5.8472% as compared to without the use of any data augmentation technique and 4.1009% as compared to the existing data augmentation-based CNN model, respectively. The main reason behind this improvement is that the impact of the over-fitting has been significantly reduced due to the major improvement in the size of the training dataset.

In near future, the proposed model will be improved further by designing an evolving DGCNN model. Additionally, the proposed DGCNN model will be applied on other datasets too.

Declarations

Conflict of interest The authors have declared that no conflict of interests or competing interests exist.

References

1. Acar E, Sahin E, Yilmaz I (2021) Improving effectiveness of different deep learning-based models for detecting COVID-19 from computed tomography (CT) images. *Neural Comput & Applic* 33:17589–17609
2. Al-Shargabi AA, Alshobaili JF, Alabdulatif A, Alrobah N (2021) COVID-CGAN: efficient deep learning approach for COVID-19 detection based on CXR images using conditional GANs. *Appl Sci* 11:7174
3. Apostolopoulos ID, Bessiana T, “COVID-19: Automatic Detection from X-Ray Images Utilizing Transfer Learning with Convolutional Neural Networks”, arXiv:2003.11617

4. Beers A, Brown JM, Chang K, Campbell JP, Ostmo S, Chiang MF, and Kalpathy-Cramer J (2018) “High-resolution medical image synthesis using progressively grown generative adversarial networks,” ArXiv, vol. abs/1805.03144.
5. Chen H, Cao P Deep learning-based data augmentation and classification for limited medical data learning. In: 2019 IEEE international conference on power, intelligent computing and systems (ICPICS) 2019 Jul 12. IEEE, pp 300–303
6. Chen X, Xu J, Zhou R, Chen W, Fang J, Liu C (2021) TrajVAE: a Variational AutoEncoder model for trajectory generation. *Neurocomputing*. 428:332–339
7. Dataset_4_classes, Transfer-Learning-COVID-19. [Accessed: July 24, 2021], [Available Online]: <https://github.com/vj2050/Transfer-Learning-COVID-19>
8. de la Iglesia Vayá M, Saborit JM, Montell JA, Pertusa A, Bustos A, Cazorla M, Galant J, Barber X, Orozco-Beltrán D, García F, et al. (2020) Bimcv COVID-19+: a large annotated dataset of rx and ct images from COVID-19 patients. arXiv preprint arXiv:2006.01174
9. DeGrave AJ, Janizek JD, and Lee S-I (2020) Ai for radiographic COVID-19 detection selects shortcuts over the signal. medRxiv,
10. Erol B, Gurbuz SZ, Amin MG (2020) Motion classification using kinematically sifted acgan-synthesized radar micro-doppler signatures. *IEEE Trans Aerosp Electron Syst* 56(4):3197–3213
11. Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H (2018) GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*. 321:321–331
12. Gonog L, Zhou Y (2019) A review: generative adversarial networks. In: 2019 14th IEEE conference on industrial electronics and applications (ICIEA). IEEE, pp 505–510
13. Hemdan EED, Shouman MA, Karar ME (2020) COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images. 2020 arXiv preprint arXiv:2003.11055
14. Motamed S, Rogalla P, Khalvati F (2021) RANDGAN: randomized generative adversarial network for detection of COVID-19 in chest X-ray. *Sci Rep* 11(1):1
15. Narin A, Kaya C, Pamuk Z (2020) “Automatic Detection of Coronavirus Disease (COVID-19) Using X-Ray Images and Deep Convolutional Neural Networks” arXiv preprint arXiv:2003.10849
16. Ng M-Y, Lee EYP, Yang J, Yang F, Li X, Wang H, Lui MM-S, Lo CS-Y, Leung B, Khong P-L et al (2020) Imaging profile of the COVID-19 infection: radiologic findings and literature review. *Radiol: Cardiothoracic Imag* 2(1):e200034
17. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv, 2016; arXiv:1511.06434
18. Rubin M, Stein O, Turko NA, Nygate Y, Roitshtain D, Karako L, Barnea I, Giryres R, Shaked NT (2019) TOP-GAN: stain-free cancer cell classification using deep learning with a small training set. *Med Image Anal* 57:176–185
19. Schlegl T, Seeßböck P, Waldstein SM, Schmidt-Erfurth U, and Langs G (2017) “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in IPMI
20. Sethy PK, Behera SK (2020) Detection of coronavirus disease (COVID-19) based on deep features
21. Sharma A, Rani S, Gupta D (2020) Artificial intelligence-based classification of chest X-ray images into COVID-19 and other infectious diseases. *Int J Biomed Imaging* 6:2020
22. Shi G, Wang J, Qiang Y, Yang X, Zhao J, Hao R, Yang W, Du Q, Kazihise NG (2020) Knowledge-guided synthetic medical image adversarial augmentation for ultrasonography thyroid nodule classification. *Comput Methods Prog Biomed* 196:105611
23. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) COVIDgan: data augmentation using auxiliary classifier Gan for improved COVID-19 detection. *Ieee Access* 8:91916–91923
24. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) CovidGAN: data augmentation using auxiliary classifier GAN for improved Covid-19 detection. *IEEE Access* 8:91916–91923
25. Wang L, Wong A (2020) “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images”. arXiv preprint arXiv:2003.09871.
26. Wang L and Wong Alexander (2020) COVID-net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. arXiv preprint arXiv:2003.09871
27. Zhao D, Zhu D, Lu J, Luo Y, Zhang G (2018) Synthetic medical images using F&BGAN for improved lung nodules classification by MultiScale VGG16. *Symmetry* 10:519