

Article

COVID-CGAN: Efficient Deep Learning Approach for COVID-19 Detection Based on CXR Images Using Conditional GANs

Amal A. Al-Shargabi ^{1,*}, Jowharah F. Alshobaili ¹, Abdulatif Alabdulatif ² and Naseem Alrobah ¹

¹ Department of Information Technology, College of Computer, Qassim University, Buraydah 51921, Saudi Arabia; j.alshobaili@qu.edu.sa (J.F.A.); 411200050@qu.edu.sa (N.A.)

² Department of Computer Science, College of Computer, Qassim University, Buraydah 51921, Saudi Arabia; ab.alabdulatif@qu.edu.sa

* Correspondence: a.alshargabi@qu.edu.sa

Abstract: COVID-19, a novel coronavirus infectious disease, has spread around the world, resulting in a large number of deaths. Due to a lack of physicians, emergency facilities, and equipment, medical systems have been unable to treat all patients in many countries. Deep learning is a promising approach for providing solutions to COVID-19 based on patients' medical images. As COVID-19 is a new disease, its related dataset is still being collected and published. Small COVID-19 datasets may not be sufficient to build powerful deep learning detection models. Such models are often over-fitted, and their prediction results cannot be generalized. To fill this gap, we propose a deep learning approach for accurately detecting COVID-19 cases based on chest X-ray (CXR) images. For the proposed approach, named COVID-CGAN, we first generated a larger dataset using generative adversarial networks (GANs). Specifically, a customized conditional GAN (CGAN) was designed to generate the target COVID-19 CXR images. The expanded dataset, which contains 84.8% generated images and 15.2% original images, was then used for training five deep detection models: InceptionResNetV2, Xception, SqueezeNet, VGG16, and AlexNet. The results show that the use of the synthetic CXR images, which were generated by the customized CGAN, helped all deep learning models to achieve high detection accuracies. In particular, the highest accuracy was achieved by the InceptionResNetV2 model, which was 99.72% accurate with only ten epochs. All five models achieved kappa coefficients between 0.81 and 1, which is interpreted as an almost perfect agreement between the actual labels and the detected labels. Furthermore, the experiment showed that some models were faster yet smaller compared to the others but could still achieve high accuracy. For instance, SqueezeNet, which is a small network, required only three minutes and achieved comparable accuracy to larger networks such as InceptionResNetV2, which needed about 143 min. Our proposed approach can be applied to other fields with scarce datasets.

Keywords: COVID-19 detection; generative adversarial network (GAN); conditional GAN (CGAN); deep learning; chest x-ray (CXR)



Citation: Al-Shargabi, A.A.; Alshobaili, J.F.; Alabdulatif, A.; Alrobah, N. COVID-CGAN: Efficient Deep Learning Approach for COVID-19 Detection Based on CXR Images Using Conditional GANs. *Appl. Sci.* **2021**, *11*, 7174. <https://doi.org/10.3390/app11167174>

Academic Editors: Panagiotis G. Asteris and Amir H. Gandomi

Received: 30 June 2021
Accepted: 2 August 2021
Published: 4 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The coronavirus (COVID-19) outbreak has resulted in a worldwide crisis; globally, the rate of cases has risen rapidly. The World Health Organization (WHO) has reported 131,020,967 confirmed COVID-19 cases and 2,850,521 deaths worldwide as of 6 April 2021 [1]. To investigate COVID-19 symptoms, a chest X-ray (CXR) imaging is considered a first-line diagnostic examination. However, medical providers may suffer from a lack of sufficient experience to conduct accurate diagnoses based on medical images such as CXR and computed tomography (CT) due to the rapid and vast changes in the COVID-19 virus. Therefore, artificial intelligence (AI), especially deep learning techniques, can be applied as powerful tools to support practitioners' efforts to automate diagnostic procedures for

COVID-19. Deep learning solutions show promise as tools for developing medical-image-based solutions for diagnosing COVID-19. Diagnosis via medical images using deep learning techniques can provide precise and concise analysis results.

One of the main concerns is the lack of available medical image datasets related to COVID-19, such as CXR and CT scans; the available datasets are considered small and insufficient in terms of their size. This was expected, as COVID-19 is new to research, which has resulted in the collection of small, relevant, labeled datasets and a lack of standardized data. This is a major challenge for diagnosing COVID-19 using deep learning models, where the size of medical image samples is insufficient for meaningful learning in the clinical practice context [2].

Although existing deep learning detection models of COVID-19 achieve high performance, these models may not be trusted as they are based on a small dataset. Overfitting is an analysis that very closely resembles a specific dataset, and therefore most fail to properly fit new unseen observations. Overfitting is a common problem that occurs with deep learning models that are trained on limited datasets.

To overcome this issue, we suggest the use of generative adversarial networks (GANs) [3]. GANs can generate new images by learning the data distribution of the original dataset. In this study, a conditional GAN (CGAN), which is a special type of the basic GAN, was designed specifically for generating COVID-19 CXR images. CGAN [4], which extends the GAN, can generate specific class data samples, for example, by manipulating an image label that is a part of dataset metadata. CGAN was chosen over the basic GAN as the quality of the sample generation process is enhanced due to the direct CGAN training process [5,6]. In particular, we propose COVID-CGAN, a detection model for COVID-19 patients using multiple deep learning models based on the CXR images generated by the designed CGAN. Several researchers [6–8] have applied GANs to synthesize COVID-19 images based on CXR or CT images. However, these studies did not create detection models to diagnose COVID-19 based on the generated images. Some other researchers constructed deep learning detection models for COVID-19, but such studies were mainly based on relatively small datasets [9–16].

The main contributions of this paper are as follows:

- A customized CGAN was designed to generate CXR images that can be used for COVID-19 detection studies. This includes the architectures of generator and discriminator networks as well as parameter configurations;
- An expanded COVID-19 CXR dataset that involves 3290 images that can be used to build COVID-19 detection models was generated;
- COVID-19 detection using the synthetic CXR images generated by the CGAN was demonstrated.

This paper is structured as follows: Section 2 presents the related work. Section 3 describes the materials and methods used in our experimental work. Section 4 presents the results, and Section 5 discusses the main findings and concludes the paper.

2. Related Work

This section provides an overview of the state-of-the-art studies in relation to COVID-19 detection based on CXR images. The section also provides a summary of previous studies suggesting that applied GANs are able to overcome the problem posed by the small COVID-19 dataset size.

Table 1 presents a summary of COVID-19-related studies in terms of image type used, dataset size, GAN architecture applied to expand the dataset in cases where GANs are used, classification method, and classification model performance.

Table 1. A summary of COVID-19-related studies.

Study	Image Type	Dataset Size (of COVID-19 Cases)	GAN Used	Classification Method	Model Performance
[9]	CXR	170	-	AlexNet	98% (Acc.)
[10]	CXR	183	-	Customized CNN model (COVID-Net)	92.6% (Acc.)
[11]	CXR	100	-	ResNet50, InceptionV3, Inception-ResNetV2	98% (Acc.)
[12]	CXR	100	-	Customized CNN model	96% (Acc.)
[13]	CXR	102	-	ResNet50 and VGG16	94.4% (Acc.)
[14]	CXR	68	-	ResNet-50	96.23% (Acc.)
[16]	CXR	145	-	ResNet34, ResNet50, DenseNet169, VGG19, Inception ResNetV2, RNN	95.72% (Acc.)
[17]	CXR	219	-	KNN, SVM, DT	98.97% (Acc.) 89.39% (Sens.) 99.75% (Spec.) 96.72% (F1-score)
[18]	CXR	120	-	NASNet	97% (Acc.) 97% (Sens.)
[19]	CXR	70	-	VGG16	99% (F1-score)
[20]	CXR	423	-	MobileNetv2, SqueezeNet, ResNet18, InceptionV3, ResNet101, CheXNet, VGG19, DenseNet201	99.7%, (Acc.) 99.7%, (Prec.) 99.7% (Sens.) 99.55% (Spec.)
[21]	CXR	127	-	SVM	95.33% (Acc.)
[22]	CXR	295	-	SVM	99.27% (Acc.)
[23]	CXR	250	-	VGG16	0.98 (Acc.) 0.89 (F-1 score) 0.94 (Spec.) 0.87 (Sens.)
[24]	CXR	184	-	ResNet18, ResNet50, SqueezeNet, DenseNet-121	98% (Sens.) 90% (Spec.)
[25]	CXR	284	-	Xception	89.6% (Acc.) 93% (Prec.) 98.2% (Sens.)
[26]	CXR	90	-	KNN, SVM, MLP, DT, RF	0.89 (F-1 score)
[27]	CXR	127	-	DarkNet	98.08% (Acc.)
[28]	CXR	132	-	VGG16	86% (Acc.) 86% (Sens.) 93% (Spec.) 90% (AUC)
[29]	CXR	180	-	Patch-based CNNs that consist of a number of ResNets	88.9% (Acc.) 84.4% (Sens.) 96.4% (Spec.)
[30]	CXR	25	-	NB, NN, SVM, RBF, KNN, SGD, LR, RF, DT, AdaBoost	A table of 12 classifiers and 10 evaluation criteria
[31]	CXR	313	-	Customized CNN, VGG16, VGG19, Inception-V3, Xception, InceptionResNet-V2, MobileNet- V2, DenseNet-201, NasNet-mobile	99.01% (Acc.) 99.72% (AUC)

Table 1. Cont.

Study	Image Type	Dataset Size (of COVID-19 Cases)	GAN Used	Classification Method	Model Performance
[6]	CXR	403	ACGAN	VGG16	95% (Acc.) 96% (Prec.) 90% (Recall)
[7]	CXR	69	Traditional GAN	AlexNet, GoogleNet, Resnet18	100% (Acc.)
[8]	CXR	337	Customized GAN (CVAE- GAN)	InceptionV3, ResNet	85–87% (Acc.)
[32]	CT	345	CGAN	AlexNet, VGGNet16, VGGNet19, GoogleNet, ResNet50	81.41% (Acc.)

Acc.: Accuracy; Sens.: Sensitivity; Prec.: Precision; Spec.: Specificity; AUC: Area under the curve.

From Table 1, we can draw the following observations:

- The majority of the studies used small datasets to feed their classification models. The maximum size of these datasets was 423 images. Some of these studies used internal datasets that have not been published for public use;
- Although most of the models achieved high accuracies, the prediction results based on them cannot be generalized due to the limited samples on which the models were trained;
- There are no studies that applied CGANs to CXR. To the best of our knowledge, one study applied a CGAN to CT but did not achieve more than 81.41% accuracy.

The studies shown in Table 1 applied several machine learning methods such as support vector machines (SVM) and random forest (RF) as well as deep learning methods such as AlexNet, GoogleNet, SqueezeNet, and ResNet18. Most of these methods achieved high performance in detecting COVID-19 cases. Moreover, only a few studies [6–8,17] applied GANs to expand the COVID-19 datasets. Specifically, these studies applied traditional GANs or other GAN architectures, including CGAN, auxiliary classifier GAN (ACGAN), and conditional variational auto-encoder GAN (CVAE-GAN).

In the following, we provide a summary of the related studies. Maghdid et al. [9] developed a simple convolutional neural network (CNN) and a modified AlexNet model to classify CXR and CT scan images. The results of the study showed that the two applied models provided accuracies of 94.1% and 98%, respectively. Wang and Wong [10] introduced a customized convolutional neural network, i.e., COVID-Net, for COVID-19 cases diagnosis and made the used dataset available for public use. The model achieved an accuracy of 92.4%.

Narin et al. [11] proposed three different convolutional network models to detect COVID-19 patients, namely, ResNet50, InceptionV3, and Inception-ResNetV2. The results showed that the ResNet50 model obtained the highest recognition performance, with 98% accuracy, compared to the other two proposed models.

Zhang et al. [12] used 100 CXR images of COVID-19 patients and another 1431 CXR images of other pneumonia cases. Based on this dataset, the authors developed a new deep learning model for detection, which achieved a sensitivity of 96%. Hall et al. [13] used pre-trained Resnet50 and VGG16 plus their own small CNN trained on 102 COVID-19 cases and 102 other pneumonia cases in 10-fold cross-validation. Their model achieved an accuracy of 94.4%.

Farooq and Hafeez [14] proposed a public dataset for COVID-19 cases and other pneumonia cases to be used by the public. The authors also proposed a novel model, i.e., COVID-ResNet, that was tested to reduce training time. The model achieved an accuracy of 96.23%.

Hammoudi et al. [16] proposed a number of deep learning models, i.e., ResNet34, ResNet50, DenseNet169, VGG-19, Inception ResNetV2, and RNN, to detect pneumonia infection cases, notably viral cases. They claimed that cases of viral pneumonia diagnosed during the COVID-19 period are extremely likely to be COVID-19 infections. The DenseNet169 architecture reached the best performance with an average classification accuracy of 95.72% on the CXR Images Pneumonia dataset.

Nour et al. [17] proposed a CNN-based hybrid model for detecting COVID-19. The models used CNN as a feature extractor, and the extracted features were applied as the input to classical machine learning techniques, specifically k-nearest neighbor, support vector machine (SVM), and decision tree, to perform the classification task. The CNN-SVM model achieved 98.97% accuracy, 89.39% sensitivity, 99.75% specificity, and 96.72% F1-score. Neural architecture search network (NASNet) is a deep convolutional network architecture developed by the Google Brain team in 2018. NASNet effectively shows improved performance on classification tasks.

Martínez et al. [18] proposed the COVID-19 detection model based on NASNet. The model achieved 97% in both accuracy and sensitivity. Alazab et al. [19] proposed a model based on VGG-16 that detects COVID-19 from CXR images. The model achieved a 99% F1 score. They also proposed a model that predicted the cases of COVID-19 confirmations, recoveries, and deaths over the next seven days in Jordan and Australia using three architectures: prophet algorithm (PA), the autoregressive integrated moving average (ARIMA) model, and long short-term memory neural network (LSTM). The empirical results showed that PA achieved an average accuracy of 94.8 and 88.43% in predicting the cases in Australia and Jordan, respectively.

To overcome the scarcity of data related to COVID-19 symptoms, Chowdhury et al. [20] used transfer learning techniques and data augmentation to train pre-trained deep CNN models to detect COVID-19 from CXR images. They used CheXNet, a CNN model that consists of 121 layers trained on CXR images to detect different respiratory diseases, in addition to MobileNetv2, SqueezeNet, ResNet18, InceptionV3, ResNet101, VGG19, and DenseNet201. The empirical evaluation showed that all the models achieved high performance in detecting COVID-19, and the highest performance was achieved by both CheXNet and DenseNet201 when the data augmentation technique was applied. Sethy et al. [21] introduced two types of hybrid models that use SVM as a classifier in detecting COVID-19 based on CXR images. Each model of the first type consists of a deep pre-trained CNN model, specifically AlexNet, VGG16, VGG19, GoogleNet, ResNet18, ResNet50, ResNet101, InceptionV3, InceptionResNetV2, DenseNet201, XceptionNet, MobileNetV2, and ShuffleNet, as the feature extractor, and the extracted features were then fed to SVM to perform the classification task. The second type used traditional image classification methods, e.g., local binary patterns (LBP), a histogram of oriented gradients (HOG), and a gray level co-occurrence matrix (GLCM), with an SVM classifier. The experiment showed that the ResNet50-based hybrid model achieved the best performance with an accuracy of 95.33%.

To determine the effect of preprocessing on improving the detection of COVID-19 in CXR images, Togaçar et al. [22] focused on preprocessing the images using fuzzy and stacking techniques. To evaluate the preprocessing step's effectiveness, two deep learning models, MobileNetV2 and SqueezeNet, with SVM as a classifier, were trained with the original dataset and the preprocessed dataset. The experiment results proved that the preprocessing step improves the classification models.

Brunese et al. [23] introduced a three-stage architecture that detects and highlights the lung area infected by COVID-19 from CXR images. The model uses VGG-16 in the first and second stages to differentiate between healthy and COVID-19 cases. If the image is classified as COVID-19, the gradient-weighted class activation mapping (Grad-CAM) algorithm is used to generate an activation map that visually illustrates the infected areas. Minaee et al. [24] prepared a dataset of 5000 CXR images, called COVID-Xray-5k, that includes 184 CXR images of COVID-19. They compared the performance of four pre-trained convolutional models trained on the COVID-Xray-5k dataset, specifically ResNet18,

ResNet50, SqueezeNet, and DenseNet-121. Khan et al. [25] proposed a deep convolutional neural network model, called CoroNet, based on Xception, that is able to distinguish between three types of pneumonia: viral pneumonia, bacterial pneumonia, and COVID-19, from CXR images. Pereira et al. [26] comparatively studied different feature extraction approaches with different classification algorithms. To train the models, they prepared a dataset of CXR images called RYDLS-20 with an unbalanced distribution of the classes to reflect the real-world distribution. The dataset has CXR images that belong to normal, majority, and six other pneumonia classes, including COVID-19. Ozturk et al. [27] proposed DarkCovidNet, a deep learning model based on the DarkNet model that works as a binary classifier and multi-class classifier to classify CXR images into COVID or no findings, and COVID, pneumonia, or no findings. Civit-Masot et al. [28] proposed a multi-class classification model that uses VGG-16 to classify CXR images into COVID-19, pneumonia, or healthy.

To effectively train a deep network for detecting COVID-19 with a limited dataset, Oh et al. [29] introduced a patch-based convolutional neural network algorithm consisting of a segmentation network and a classification network. The model extracts the lung area from the CXR image, and different patches of the extracted area are classified with a classification network that consists of a network of ResNet. Mohammed et al., in their benchmarking paper [30], stated that the large number of machine learning models proposed for detecting COVID-19 from CXR images complicates the task for health organizations in selecting an appropriate model. Therefore, they proposed a benchmarking methodology using a multi-criteria decision-making (MCDM) model that evaluated 12 machine learning classification algorithms on ten time-consuming and accuracy parameters, e.g., precision and recall. To conduct the experiment, they constructed a dataset of 50 CXR images, including 25 COVID-19 CXR images, and the InceptionV3 model was used as a feature extractor. Rajaraman et al. in [31] evaluated the performance of a custom CNN and eight different pre-trained CNN-based models, e.g., VGG-16 and Inception-V3, in detecting COVID-19 from CXR images. To reduce model complexity, and improve robustness and generalization, they used modality-specific transfer learning, iterative pruning, and ensemble strategies.

Abdul Waheed et al. [6] introduced an ACGAN-based model for generating CXR images of COVID-19. They compared the performance of a VGG16 model trained on the original dataset to another VGG16 model trained on a dataset augmented by the ACGAN model. Based on CT scan images, Loey et al. [32] applied CGANs to differentiate COVID-19 and non-COVID-19 images.

In this study, we propose using CGANs to generate COVID-19 CXR images and then using them along with the original images to detect COVID-19 cases. Although Loey et al. [32] applied a CGAN, they used it with CT images, not CXR images. Loey et al. [7] applied a GAN to CXR images to detect respiratory diseases from the images. The original dataset contained 306 CXR images in four classes, 97 of which were COVID-19 CXR images. The best-performing model was GoogleNet, which achieved 100% accuracy in the discrimination of normal and COVID-19 CXR images compared to 52.8% before using the GAN. Based on 16 classes, Albahli [8] empirically evaluated the performance of several deep-learning-based models that detect and classify different respiratory diseases, including COVID-19, from CXR images. The results showed that the best-performing model was ResNet, which achieved 85 to 87% validation accuracy.

3. Materials and Methods

3.1. Overview of COVID-CGAN

Figure 1 depicts an overview of COVID-CGAN. As shown in the figure, a dataset of CXR images was collected. The dataset includes normal and COVID-19 images. To ensure that our smart detector is able to accurately recognize COVID-19 images, pneumonia images were added to the dataset. The collected dataset was augmented using generative models that synthesize images from the dataset. Specifically, a CGAN was designed to

generate the images. The expanded dataset resulting from this phase was then used to detect COVID-19 patients based on their CXR images. In particular, the generated images were used to train multiple deep learning models based on certain configurations of the model’s parameters. The trained model was later used to predict the presence of COVID-19 based on testing data.

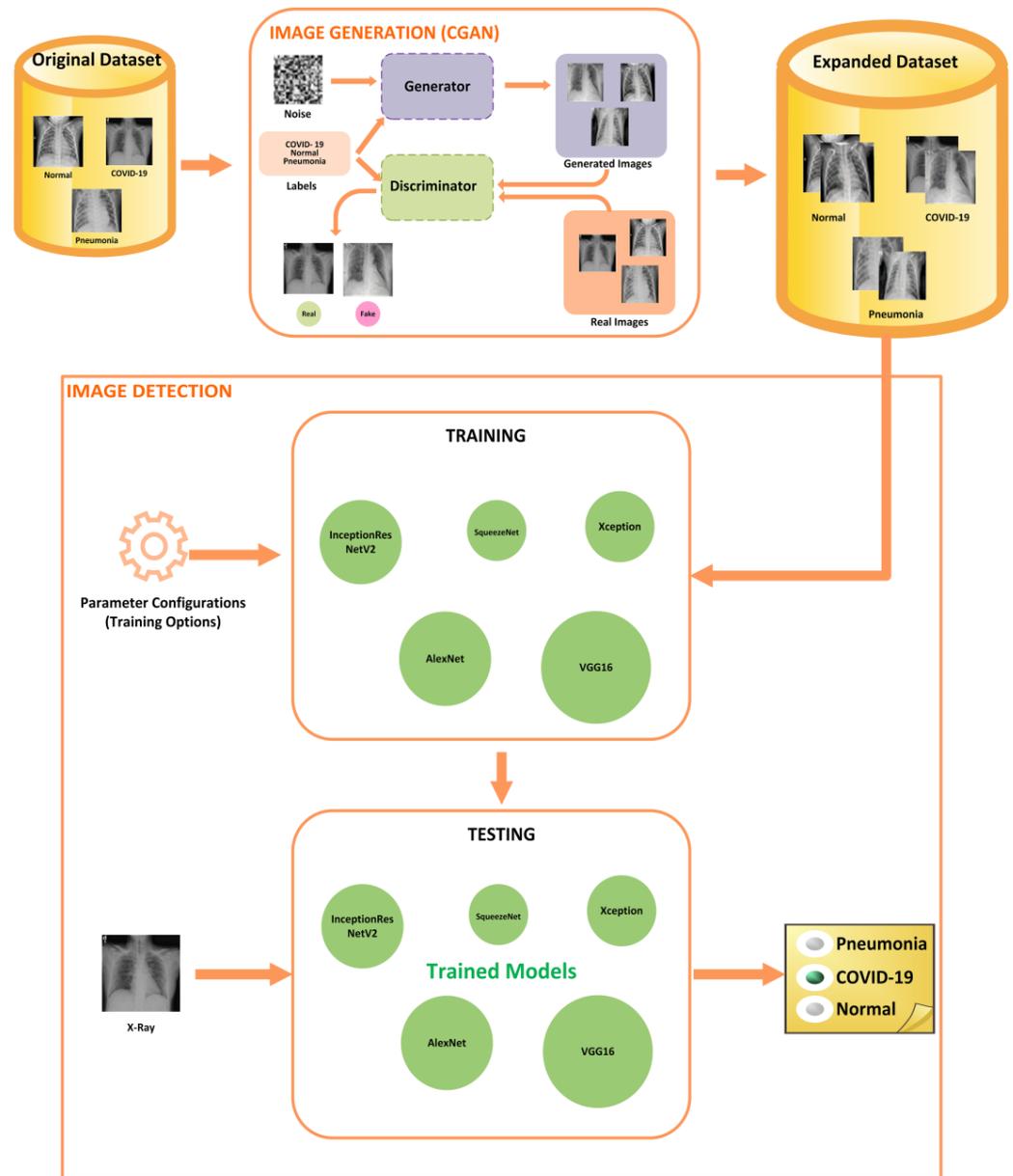


Figure 1. A block diagram of COVID-CGAN.

The details of the image generation phase, i.e., dataset expansion and the deep learning approach to COVID-19 detection, are presented in the following two subsections.

3.2. CGAN for COVID-19 CXR Image Generation

3.2.1. Conditional Generative Adversarial Networks (CGANs)

GANs are structured to train generative models. They contain two networks: the generator (*G*) and discriminator (*D*). These two networks work against each other to produce convincing yet false images. GANs generate samples from the random noise that is provided as the input to the generator. CGANs are designed to include extra information

about images (y) for both the generator (G) and the discriminator (D), which control the class of the generated output (Figure 2).

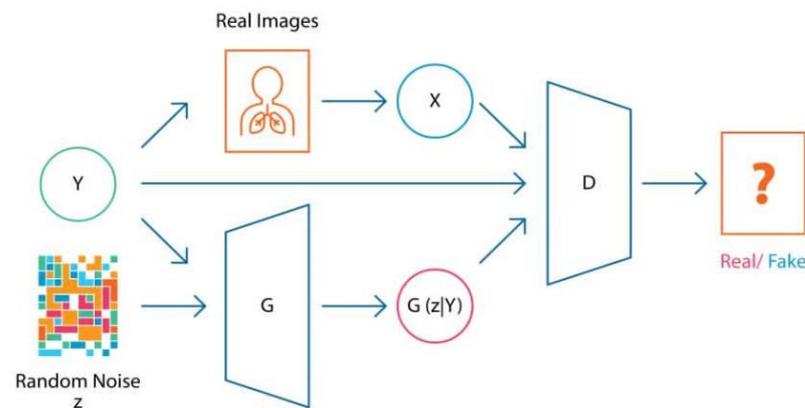


Figure 2. An overview of the CGAN architecture.

3.2.2. The Original Dataset

There are three classes in the dataset: normal, COVID-19, and pneumonia. The dataset contains around 500 CXR images for each class. As COVID-19 images are not widely available, they were collected from various datasets [33–35]. The normal and pneumonia CXR images were collected from a dataset that was published in 2018 [36], and the number of images was chosen to be balanced with the COVID-19 images. A sample of each of the three classes is shown in Figure 3. All CXR images were preprocessed to have the same color system, type, and size. Specifically, the images were unified to be RGB of PNG type. Flipping was applied to the images in which the images were reflected horizontally with 50% probability.

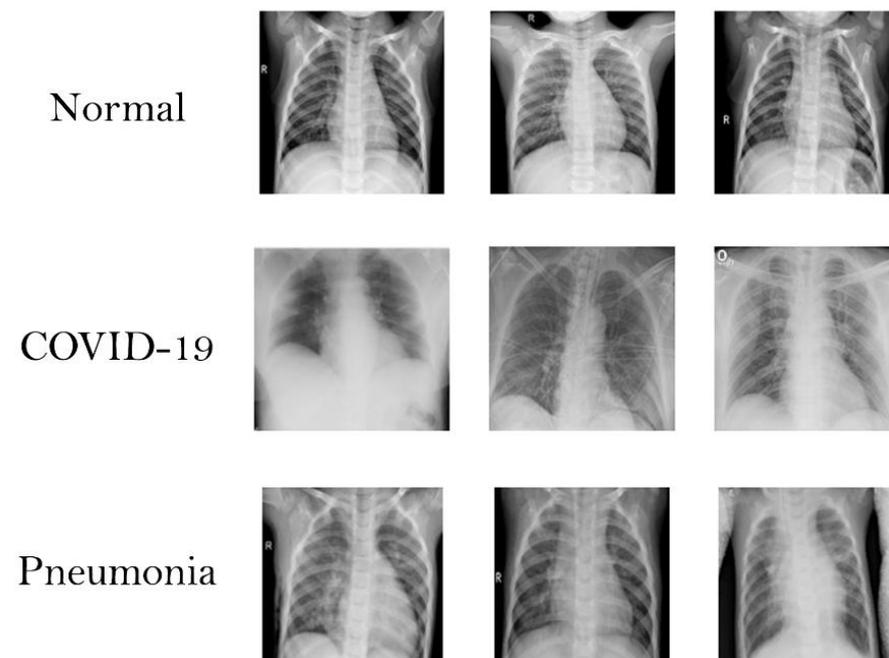


Figure 3. CXR image samples from the original dataset.

In addition, some of the images were cropped to remove any details that do not belong to the main CXR image such as the header and footer. All annotations and arrows generated by X-ray devices were also removed from the images. Additionally, all the images in the dataset were resized to be 128×128 .

3.2.3. The Proposed CGAN Architecture

Based on the theories of the CGAN presented earlier, we constructed a customized CGAN that uses CXR images from the original dataset to generate new images. The loss scores were calculated based on the following equation:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}}(x) [\log D(x|y)] + E_{z \sim P_z}(z) [\log(1 - D(G(z|y)))] \quad (1)$$

where $D(x|y)$ is the discriminator's estimate of the probability that a real data instance (x) is real for a given class (y), and $D(G(z|y))$ is the discriminator's estimate of the probability that a fake instance is real for a given class (y).

The specific architectures of the CGAN generator and discriminator networks are shown in Figures 4 and 5, respectively. As shown in Figure 4, the generator (G) contains a total of 18 layers: an input layer (noise), a project and reshape layer (proj), an embedding layer (embed), a concatenation layer (concat), five transposed convolutional layers (tconv1-tconv5), four batch normalization layers (bnorm1-bnorm4) [37], four rectified linear unit (ReLU) layers (relu1-relu4), and a hyperbolic tangent layer (tanh).

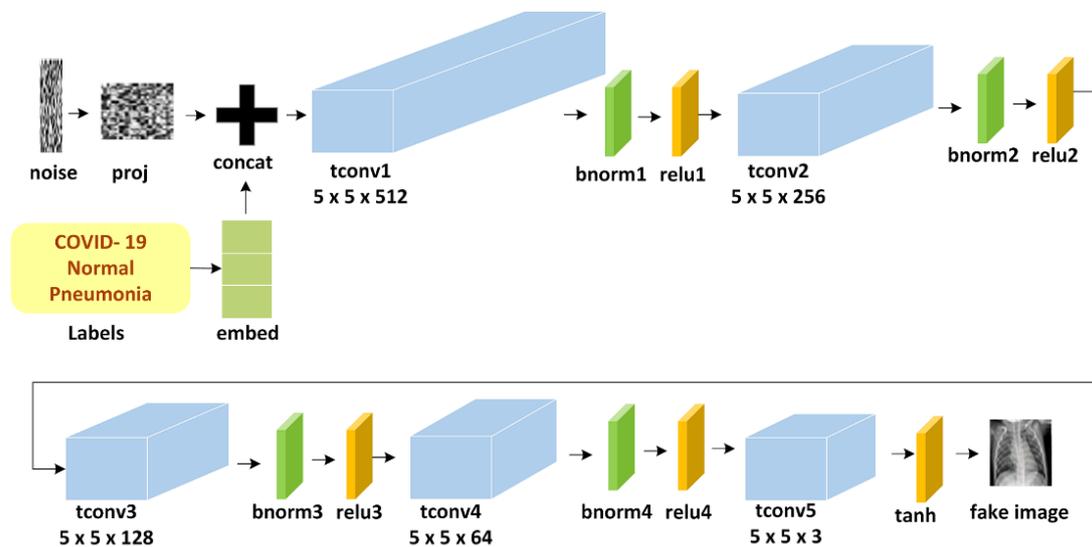


Figure 4. The proposed architecture of the CGAN generator. The generator takes a noise vector and image labels and up-samples the generated images using five transposed convolutional layers.

In particular, the network converts a noise vector of 100 samples into $4 \times 4 \times 256$ arrays using project and reshape and then up-scales the resulting arrays to $128 \times 128 \times 3$ using a series of transposed convolution layers with batch normalization and ReLU layers.

Notably, the number of the transposed convolutional neural networks as well as the filter number and sizes, i.e., 5, were chosen carefully so that the generated images were 128×128 . If the desired size is different, then the number of the transposed convolutional neural networks will differ as well. For example, if the desired size is 64×64 , then there will be four transposed convolutional neural networks instead of five.

The network converts a noise vector of 100 samples into $4 \times 4 \times 512$ arrays using the project and reshape layer. Then, it converts the categorical labels to embedding vectors and reshapes them to a 4×4 array. After, the network concatenates the resulting images from the two inputs along the channel dimension to a $4 \times 4 \times 512$ array. Next, it up-scales the resulting arrays to $128 \times 128 \times 3$ using a series of transposed convolutional layers with batch normalization and ReLU layers.

The discriminator (D), as shown in Figure 5, contains 24 layers: an input layer (in), an embedding layer (embed), a concatenation layer (concat), six convolutional layers

(tconv1–tconv6), six dropout layers (drop, drop1–drop5) [38], four batch normalization layers (bnorm2–bnorm5), and five leaky rectified linear unit layers (lrelu1–lrelu5).

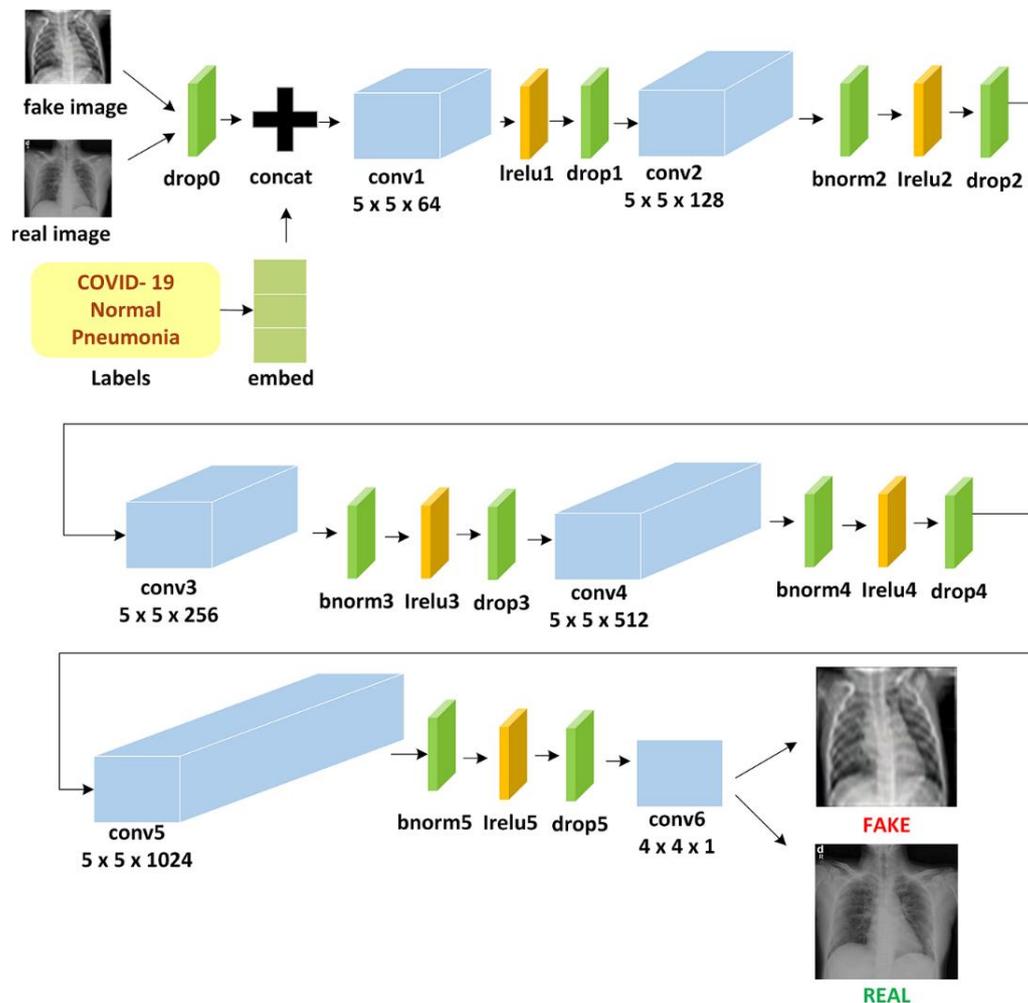


Figure 5. The proposed architecture of the CGAN discriminator. The discriminator learns from real images as well as image labels and down-samples the images produced by the generator using six convolutional layers to obtain the classification.

The addition of the dropout layers in the discriminator networks helps to reduce the capacity of the network during training and to avoid overfitting. The CGAN was trained with the original dataset described in Section 3.2.2 using the configurations shown in Table 2. The number of epochs was chosen based on the experiment so that we obtained reasonable image quality. To increase the stability of the CGAN, the generator and discriminator learning rates were not equal.

Table 2. Parameter configuration of the CGAN network.

Number of Epochs	2000
Mini Batch Size	64
Optimizer	Adam [39]
Learning rate of Generator	0.0002
Learning rate of Discriminator	0.0001

To validate the images generated by the CGAN, the train on synthetic, test on real (TSTR) method [40] was applied. In this method, the generated images are used to train a

deep learning model, VGG16, and then the model is tested on the real images. The training dataset contains the CXR images of the three classes: COVID-19, pneumonia, and normal. The training parameters' configurations used in the training process are as follows:

- Number of epochs: 10;
- Optimizer: stochastic gradient descent model (SGDM);
- Batch size: 16;
- Learning rate: 0.001.

To measure the performance of the VGG16 model, recall, precision, and F1-score metrics were used. The full descriptions of the model as well as the performance measures are available in Sections 3.3.1 and 3.3.2, respectively.

To ensure that CGAN generated distinct images, the mean squared error (MSE) and structural similarity index measure (SSIM) [41] were used. The two metrics were calculated using the following two equations:

$$\text{MSE} = \frac{1}{MN} \sum_{n=1}^M \sum_{m=1}^N [\hat{g}(n, m) - g(n, m)]^2 \quad (2)$$

where $\hat{g}(n, m)$ and $g(n, m)$ represent image1 and image2, respectively.

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3)$$

where μ and σ denote the average and standard deviation of the original images X and the test image Y, respectively; σ_{xy} is the covariance of X and Y; and C_1 and C_2 are constants that prevent numerical instabilities.

3.3. Deep Learning Models for Detecting COVID-19 Based on CXR Images

3.3.1. The Deep Learning Models

To detect COVID-19 based on CXR images, five deep transfer learning models were used: InceptionResNetV2, Xception, SqueezeNet, VGG16, and AlexNet. These models were selected to range in size, i.e., the number of layers and size on disk. In the following, we provide a brief description of each model.

- InceptionResNetV2 is a type of convolutional neural network that consists of 164 layers deep with image input size 299×299 . The architecture of InceptionResNetV2 is formulated based on a combination of the Inception structure and a residual network (ResNet) connection. The usage of a ResNet connection not only eliminates degradation issues during deep structure but also reduces the training time. The InceptionResNetV2 architecture consists of a stem block that contains three standard convolutional layers and two 3×3 max-pooling layers. Multiple convolutional and max-pooling layers follow stem blocks with different sizes and different orders using ReLU and SoftMax functions [42]. The InceptionResNetV2 architecture is depicted in Figure 6a.
- Xception is a convolutional neural network that was adapted from the Inception network, where the Inception modules are replaced with depthwise separable convolutions. The network has an image input size of 299×299 and is 71 layers deep. Figure 6b shows the architecture of Xception, which consists of multiple convolutions with 1×1 size and depthwise separable convolutions with 3×3 size using the batch normalization, ReLU, and SoftMax functions [43].
- SqueezeNet is a small convolutional neural network that is 18 layers deep. It was designed to reduce the number of parameters to fit into computer memory or be easily transmitted over computer networks. SqueezeNet begins with a standard convolutional layer followed by eight fire modules, ending with a final convolutional layer and the SoftMax function. It performs max-pooling after the first standard convolutional

layer, Fire4, Fire8, and the last standard convolutional layer [44]. Figure 6c shows the architecture of SqueezeNet.

- VGG16: The most straightforward method to improve deep neural networks' performance is by increasing the network's size. For this reason, the visual geometry group (VGG) was created with three fully connected layers, 13 convolutional layers, and smaller size filters (2×2 and 3×3) using ReLU and SoftMax functions. It performs max-pooling twice with size 2×2 [45]. The architecture of VGG16 is depicted in Figure 6d.
- AlexNet is a convolutional neural network eight layers deep. It contains five convolution layers, three max-pooling layers, and three fully-connected layers using ReLU and SoftMax functions. The input image size is 227×227 [46]. Figure 6e shows the architecture of SqueezeNet.

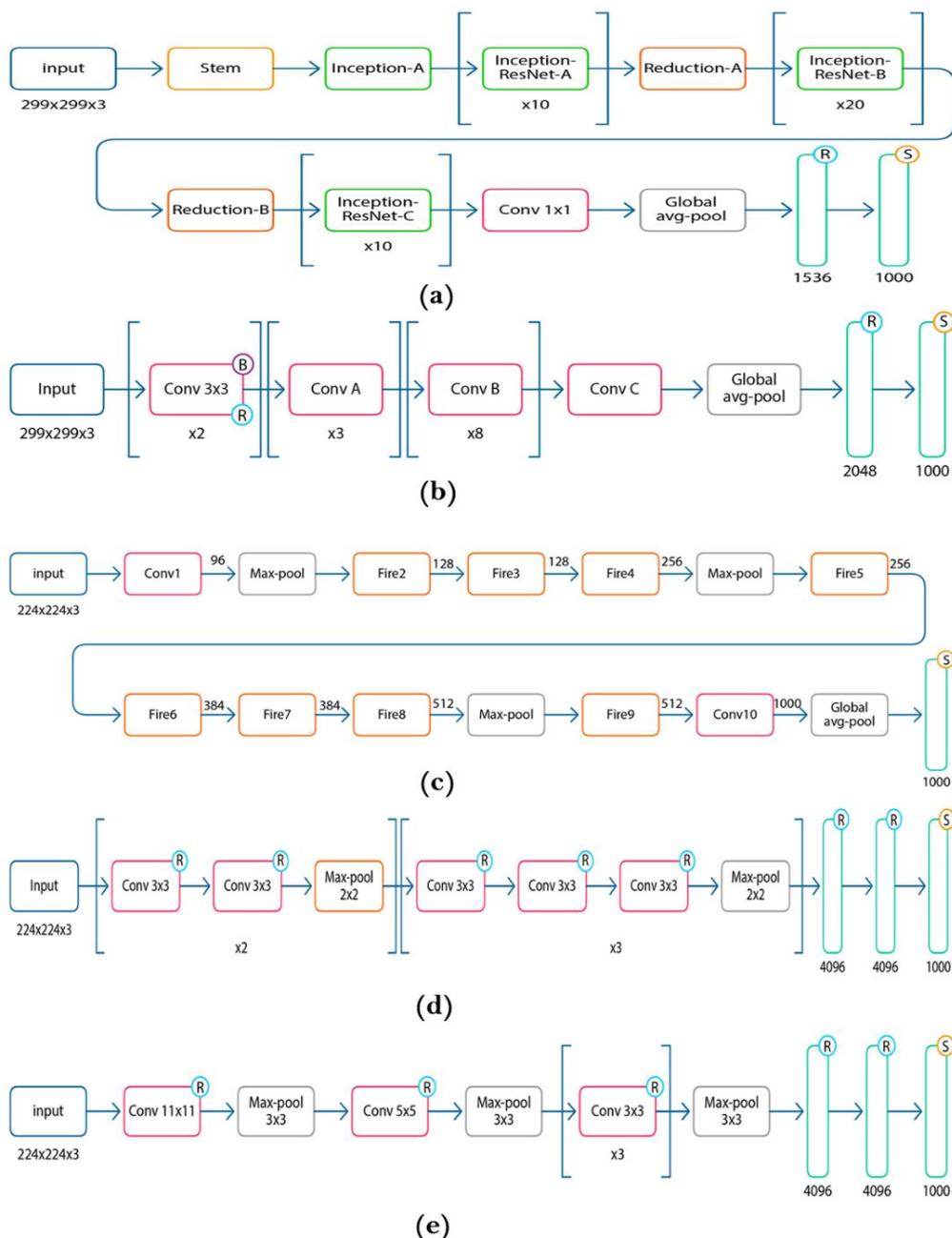


Figure 6. Deep transfer learning models used for detection: (a) Inception-ResNetV2 architecture; (b) Xception architecture; (c) SqueezeNet architecture; (d) VGG16 architecture; (e) AlexNet architecture.

The five models were trained on the expanded dataset, which contains the real images from the original dataset and the images generated by the proposed CGAN. To enable fair comparison among the different models, all the models were trained with the same parameters' configurations. Table 3 shows the parameter configurations of the training process.

Table 3. Parameter configurations used in the training process.

Name	Batch Size	Number of Epochs	Optimizer	Learning Rate
InceptionResNetV2	16	10	Stochastic gradient descent model (SGDM)	0.001
Xception	16			
SqueezeNet	64			
Vgg16	16			
AlexNet	128			

3.3.2. Performance Metrics

The most common metrics used for evaluating deep learning models are accuracy, precision (specificity), recall (sensitivity), and F1-score [47]. In addition, macro-recall, macro-precision, and macro-F are preferred for evaluating multi-label models [48]. The kappa coefficient is used to determine the degree of agreement between the reference data and the classified map. It is used to control only those instances that may have been correctly classified by chance [49]. Accordingly, these metrics were chosen for use in this study. All the metrics are based on the numbers of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) cases. In addition, the confusion matrix [50] is used to present the multi-class classification. The formulas of the metrics mentioned above are presented below:

$$Accuracy = (TP + TN) / (TP + FN + TN + FP) \quad (4)$$

$$Precision = TP / TP + FP \quad (5)$$

$$Macro - Precision = \frac{1}{N} \sum_{i=0}^N (TP / TP + FP) \quad (6)$$

$$Recall = TP / TP + FN \quad (7)$$

$$Macro - Recall = \frac{1}{N} \sum_{i=0}^N (TP / TP + FN) \quad (8)$$

$$F1score = 2 \times ((Precision \times Recall) / (Precision + Recall)) \quad (9)$$

$$Macro - F = \frac{1}{N} \sum_{i=0}^N F1Score_i \quad (10)$$

$$RandomAccuracy = ((TN + FP) \times (TN + FN) + (FN + TP) \times (FP + TP)) / (Total \times Total) \quad (11)$$

$$Kappa = (Accuracy - RandomAccuracy) / (1 - RandomAccuracy) \quad (12)$$

where N is the number of classes.

4. Experimental Results

The experimental work for this study was implemented using the deep learning toolbox in MATLAB 2020b. All the experiments were conducted using a computer device with a GPU (NVIDIA GeForce RTX 2060 8GB). The values of the training time and the accuracy may vary when different hardware and mini-batch sizes are used. The results of the image generation and detection are presented in the following two subsections.

4.1. Image Generation Results

The CGAN model was trained on the original dataset described in Section 3.2.2, which contains three classes: COVID-19, pneumonia, and normal. Figure 7 shows the CGAN training process in terms of the loss scores of the generator and the discriminator. The training process required about 16 h. The number of epochs, i.e., 2000, was chosen based on the quality of the generated images. We started with 500 epochs and increased the number until we obtained good quality images.

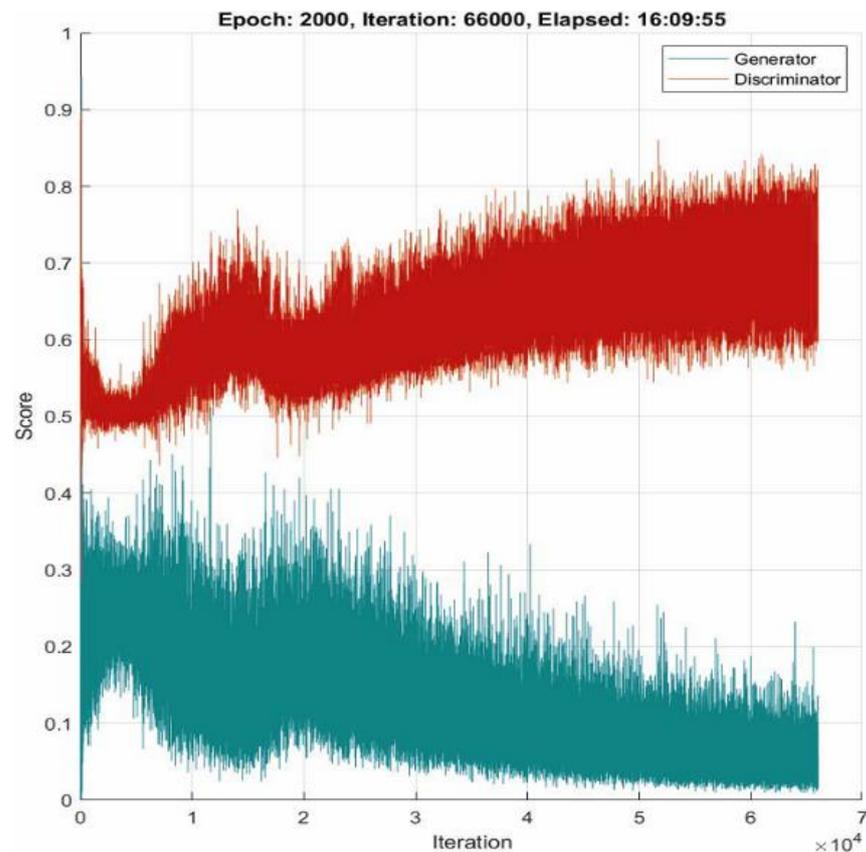


Figure 7. Generator and discriminator scores of the CGAN through the training process. The CGAN is trained on the whole dataset containing the three classes and then generates images for a given class (given label).

The generator and discriminator are competing in an adversarial manner; the generator tries to reduce the loss function to fool the discriminator, and the discriminator tries to increase the loss function to distinguish between real and fake images. Our proposed CGAN generated 2790 images for all three classes, COVID-19, pneumonia, and normal, producing 930 images for each. To obtain an overview of the images generated by the CGAN, we present samples of the three classes in Figure 8.

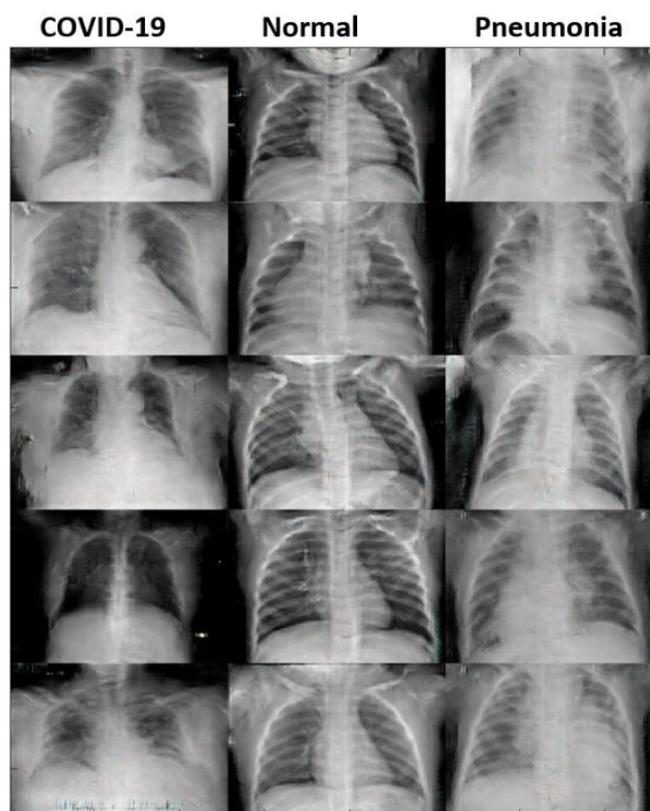


Figure 8. Sample of the generated images from the three classes by the CGAN.

To validate the images, the VGG16 model was trained on the generated images and tested on the real images only, i.e., Train on Synthetic, Test on Real (TSTR) method. Table 4 shows the performance metrics of the VGG16 for the three classes in terms of recall, precision, and F1-score. Also, the confusion matrix is shown in Figure 9. Focusing on the COVID-19 class, which is the main target of this study. The high values of the VGG16 performance measures indicate that the quality of the generated model is good enough and that the CGAN was able to produce images similar to an extent to the real images. Thus, the generated images were combined with the real images in the training and testing dataset in the following phase to perform COVID-19 detection.

An experiment was conducted: two measures were used to ensure that the generated images were distinct: MSE and SSIM. To this end, 500 images were randomly selected to calculate these two measures. A value of zero for MSE indicates perfect similarity. A value greater than one implies less similarity and will continue to grow as the average difference between pixel intensities increases as well. The SSIM value can vary between -1 and 1 , where the latter indicates perfect similarity.

Table 4. Performance metrics of image validation using VGG16.

Class	Recall	Precision	F1-Score
COVID-19	92.4	99.1	95.63
Normal	98	87.5	92.45
Pneumonia	87	91.8	89.34

True Class	COVID-19	462	7	31
	Normal	2	490	8
	PNEUMONIA	2	63	435
		COVID-19	Normal	PNEUMONIA
		Predicted Class		

Figure 9. Confusion matrix of VGG16 based on TSTR.

In total, the comparison experiment required 124,750 comparisons between these images. The average MSE and SSIM were 5593.47 and 0.22677, respectively. Since the value of MSE is greater than one, and SSIM obtained a number near zero, this means that the CGAN was able to generate distinct images. To obtain an overview of the results, Figure 10 shows samples of the MSE and SSIM values between one image and the remaining images.

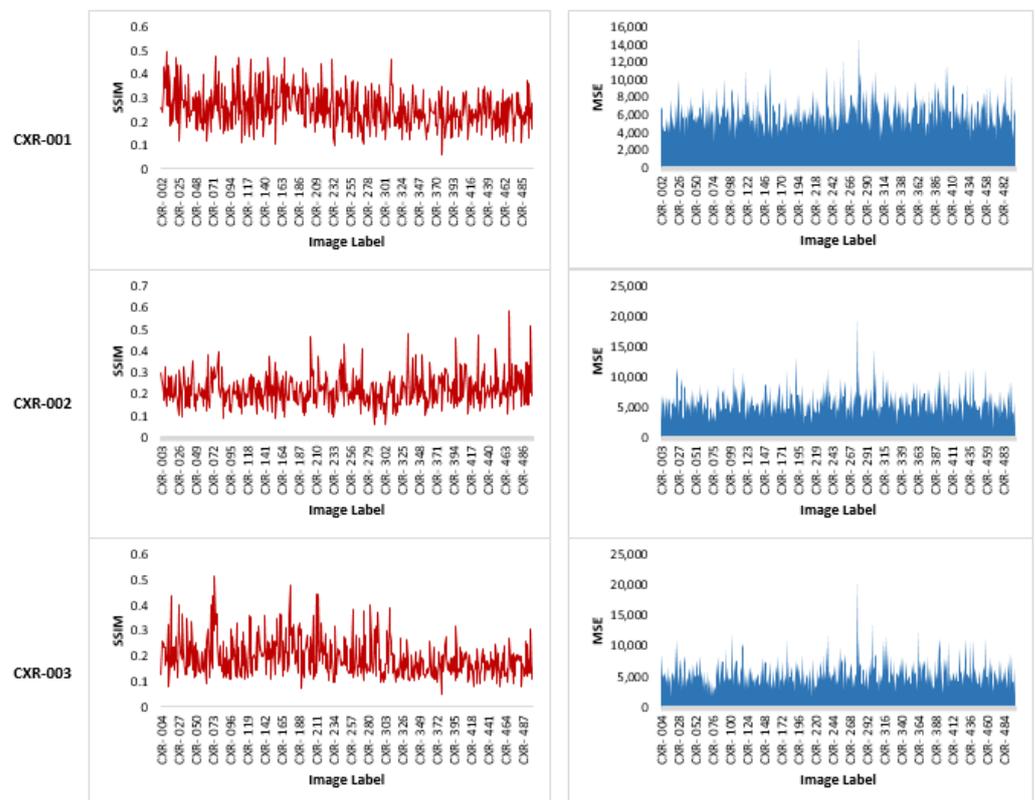


Figure 10. Samples of MSE and SSIM values between three selected images and the other images.

4.2. COVID-19 Detection Results

For detection, the deep models mentioned in Section 3.3.1 were trained and tested using the expanded datasets, which consist of 500 real images and 2790 generated images for each class. The percentage of generated images in the whole expanded dataset is 84.8%, as shown in Figure 11, which is more than five times the number of real images.

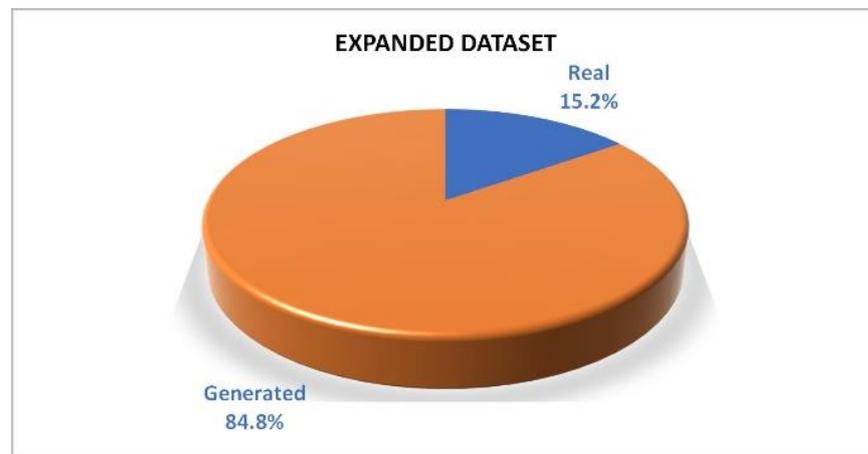


Figure 11. The percentage of real images and generated images.

Figure 12 shows the number of real and generated images in the training and testing process. As shown in the figure, the training part was conducted on 77.81% of the dataset (100 real images and 630 generated images), and the testing part was conducted on 22.19% of the dataset (400 real images and 2160 generated images). To ensure that the expanded dataset was still balanced, the same number of images was generated for each class, as shown in Table 5.

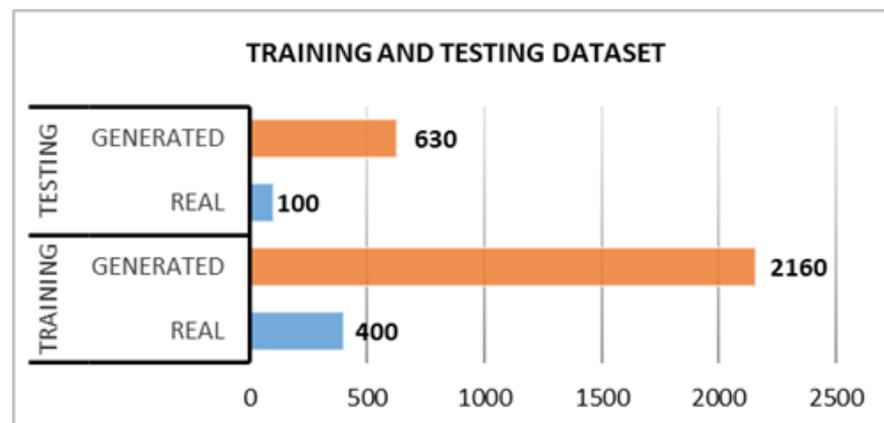


Figure 12. The number of real images and generated images in the training and testing process.

Table 5. The number of images for the training and testing process for each class.

Class	Training		Testing	
	No. of Real Images	No. of Generated Images	No. of Real Images	No. of Generated Images
COVID-19	400	2160	100	630
Normal	400	2160	100	630
Pneumonia	400	2160	100	630

The kappa coefficient, which is one of the classifier’s performance metrics, was used to measure the degree of agreement between the actual and predicted labels and to control for instances where an image may have been correctly classified by chance [49]. The kappa coefficient can be calculated using both the observed accuracy and the random accuracy as in Equations (11) and (12). According to [51], the kappa coefficient, which is between 0.81 and 1, indicates an almost perfect agreement.

Figure 13 shows that all five models obtained an almost perfect agreement between the actual labels and the predicted labels. Table 6 presents the performance of the five detection models based on accuracy, macro-precision, macro-recall, and macro-average F-score. As shown in the table, InceptionResNetV2 was the best-performing model for all metrics, whereas SqueezeNet was the worst-performing model with no apparent difference among all metrics. To obtain a better overview of the testing accuracy per class, we present the confusion matrices of the five deep transfer learning models in Figure 14.

Choosing a suitable model is a tradeoff among different parameters such as speed, size, and accuracy. Figure 15 shows the five selected models and their corresponding parameters. The area of diamonds in the figure represents the model’s size on the disk, meaning that VGG16 is the largest model with a size of 515 MB, whereas SqueezeNet is the smallest at only 4.5 MB. All models achieved high accuracy between 98.86% and 99.72%, but with different training times. The figure also shows that some models were faster or smaller compared to others but could still achieve high accuracy. For instance, SqueezeNet, which is a small network, required only three minutes and achieved comparable accuracy to larger networks such as InceptionResNetV2, which needed more time (143 min) to run.

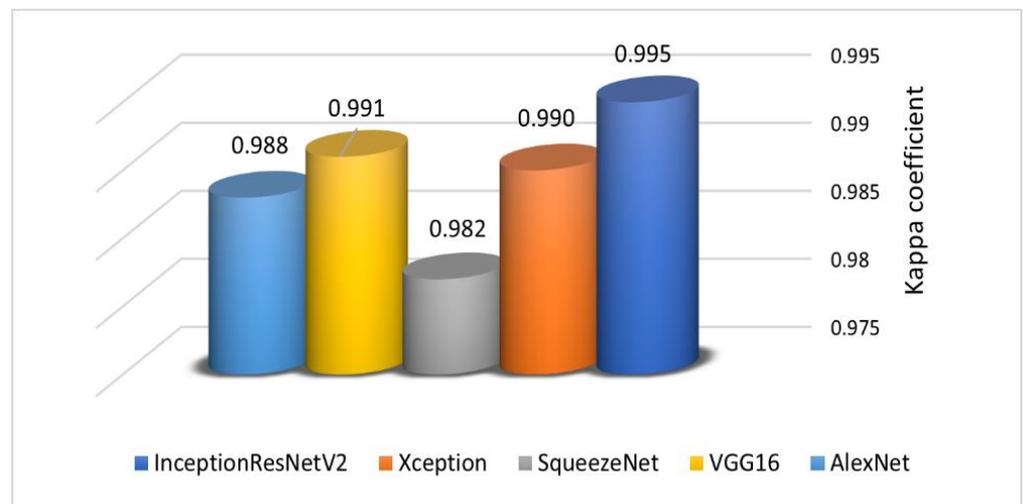


Figure 13. The number of real and generated images in the training and testing process.

Table 6. Performance metrics of the detection models.

Model	Accuracy	Macro Precision	Macro Recall	Macro F-Score
InceptionResNetV2	99.72	99.73	99.73	99.73
Xception	99.36	99.33	99.37	99.35
SqueezeNet	98.86	98.87	98.83	98.84
VGG16	99.4	99.40	99.40	99.40
AlexNet	99.32	99.23	99.23	99.23

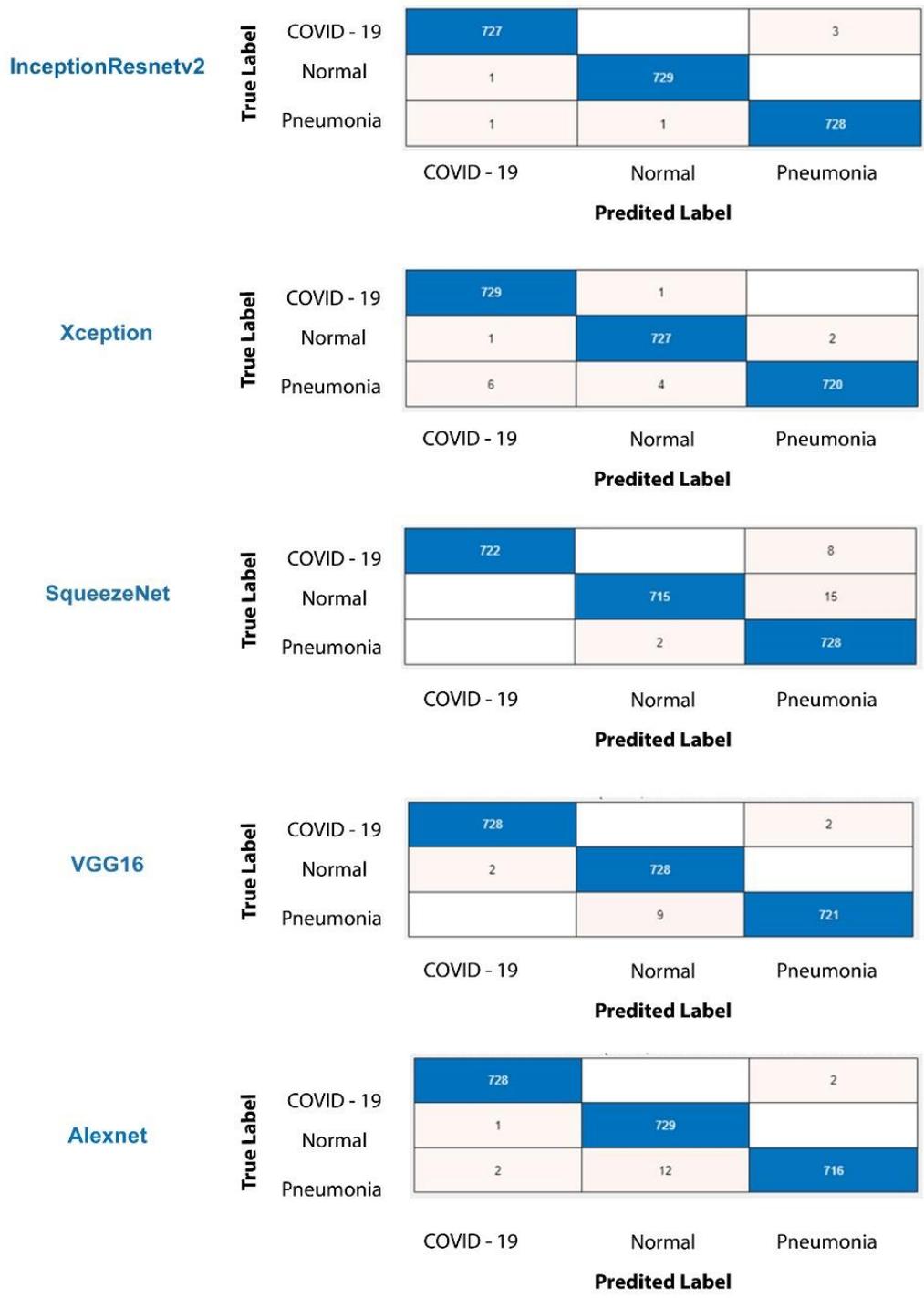


Figure 14. The confusion matrices of the deep transfer learning models.

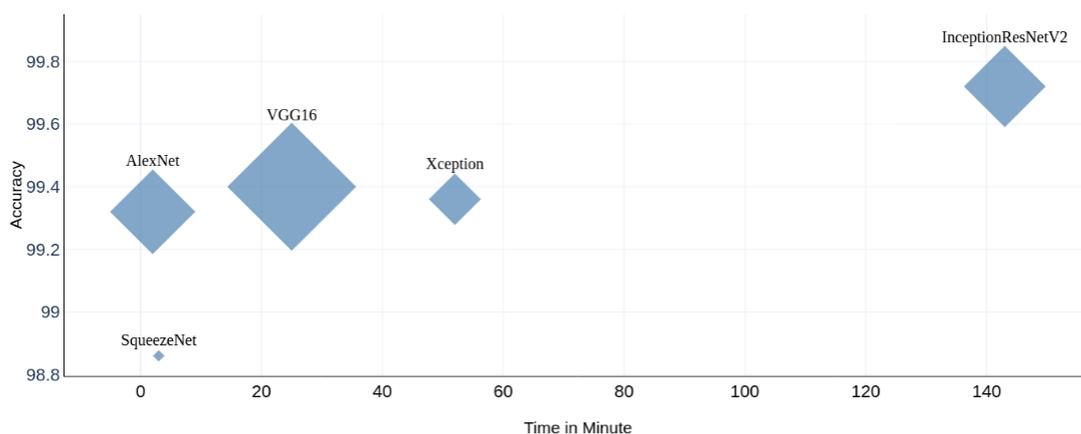


Figure 15. Classification accuracy versus training time for the five deep transfer learning models. InceptionResNetV2 achieved the highest accuracy but consumed the longest time. SqueezeNet and AlexNet were faster and achieved comparable accuracies.

5. Conclusions and Future Work

COVID-19 remains a serious disease all around the world. To improve COVID-19 detection, we proposed COVID-CGAN, an approach to extend the existing small CXR COVID-19 datasets using a CGAN and to detect the disease based on the extended datasets. We proposed a customized design for the CGAN with 18 layers for the generator and 24 layers for the discriminator.

Based on the results presented in the previous section, we derived the following findings:

1. A CGAN has a simple and straightforward architecture, yet can produce images similar to real ones. Compared to other GAN architectures that may produce better quality images, such as the least-squares generative adversarial network (LSGAN) [52] and information maximizing GAN (InfoGAN) [53], these architectures have large computational budgets and generating images is time-consuming, whereas CGANs are simpler and do not require long computation times. They can synthesize good-quality images from the original dataset.
2. Some deep learning models are better than others in terms of detecting COVID-19 patients based on their CXR images. The experimental results showed that InceptionResNetV2 outperformed other models in detecting COVID-19 based on CXR images. This model can be investigated by other researchers to detect COVID-19 based datasets other than CXR images.
3. Some deep learning models are small in size and thus provide fast predictions, yet can achieve good results in detecting COVID-19. As shown in Figure 15, SqueezeNet, which is a small network, required only three minutes to achieve accuracy comparable to that of larger networks.

Based on the present study, we suggest the following directions for future research:

- Design generative model architectures other than CGANs and compare them in terms of their ability to synthesize high-quality images that are similar to real images.
- Include patient information related to COVID-19 other than CXR, such as symptom datasets, in the diagnostic process.

Author Contributions: Conceptualization, A.A.A.-S. and J.F.A.; methodology, A.A.A.-S.; software, A.A.A.-S. and J.F.A.; validation, A.A.A.-S. and J.F.A.; investigation, A.A.; resources, J.F.A.; data curation, N.A.; writing—original draft preparation, A.A.A.-S. and J.F.A.; writing—review and editing, A.A.; visualization, J.F.A.; project administration, A.A.A.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by Qassim University, Saudi Arabia.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, for the financial support for this research under the number (coc-2020-1-1-L-10051) during the academic year 1441 AH/2020 AD.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. WHO COVID-19 Dashboard. Available online: <https://covid19.who.int/> (accessed on 21 April 2020).
2. Gozes, O.; Frid-Adar, M.; Greenspan, H.; Browning, P.D.; Zhang, H.; Ji, W.; Bernheim, A.; Siegel, E. Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring Using Deep Learning CT Image Analysis. *arXiv preprint* **2020**, arXiv:2003.05037.
3. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Proc. Syst.* **2014**, *27*, 2672–2680.
4. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv preprint* **2014**, arXiv:1411.1784.
5. Jiang, Y.; Chen, H.; Loew, M.; Ko, H. COVID-19 CT Image Synthesis with a Conditional Generative Adversarial Network. *IEEE J. Biomed. Health Inform.* **2020**, *25*, 441–452. [[CrossRef](#)]
6. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Al-Turjman, F.; Pinheiro, P.R. CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. *IEEE Access* **2020**, *8*, 91916–91923. [[CrossRef](#)]
7. Loey, M.; Smarandache, F.; Khalifa, N.E.M. Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning. *Symmetry* **2020**, *12*, 651. [[CrossRef](#)]
8. Albahli, S. Efficient GAN-based Chest Radiographs (CXR) augmentation to diagnose coronavirus disease pneumonia. *Int. J. Med. Sci.* **2020**, *17*, 1439–1448. [[CrossRef](#)] [[PubMed](#)]
9. Maghdid, H.S.; Asaad, A.T.; Ghafoor, K.Z.; Sadiq, A.S.; Khurram Khan, M. Diagnosing COVID-19 Pneumonia from X-Ray and CT Images Using Deep Learning and Transfer Learning Algorithms. *arXiv preprint* **2020**, arXiv:2004.00038.
10. Gunraj, H.; Wang, L.; Wong, A. COVIDNet-CT: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases From Chest CT Images. *pFront. Front. Med.* **2020**, *7*, 608525. [[CrossRef](#)]
11. Narin, A.; Kaya, C.; Pamuk, Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Anal. Appl.* **2021**, 1–14. [[CrossRef](#)]
12. Zhang, J.; Xie, Y.; Pang, G.; Liao, Z.; Verjans, J.; Li, W.; Sun, Z.; He, J.; Li, Y.; Shen, C.; et al. Viral Pneumonia Screening on Chest X-Rays Using Confidence-Aware Anomaly Detection. *IEEE Trans. Med. Imaging* **2021**, *40*, 879–890. [[CrossRef](#)] [[PubMed](#)]
13. Hall, L.; Goldgof, D.; Paul, R.; Goldgof, G.M. Finding COVID-19 from Chest X-rays Using Deep Learning on a Small Dataset. *arXiv preprint* **2020**, arXiv:2004.02060.
14. Farooq, M.; Hafeez, A. COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs. *arXiv preprint* **2020**, arXiv:2003.14395.
15. Eldeen, N.; Khalifa, M. Detection of Coronavirus (COVID-19) Associated Pneumonia Based on Generative Adversarial Networks and a Fine-Tuned Deep Transfer Learning Model Using Chest X-ray Dataset. *arXiv preprint* **2020**, arXiv:2004.01184.
16. Hammoudi, K.; Benhabiles, H.; Melkemi, M.; Dornaika, F.; Arganda-Carreras, I.; Collard, D.; Scherpereel, A. Deep Learning on Chest X-ray Images to Detect and Evaluate Pneumonia Cases at the Era of COVID-19. *J. Med. Syst.* **2021**, *45*, 1–10. [[CrossRef](#)] [[PubMed](#)]
17. Nour, M.; Cömert, Z.; Polat, K. A Novel Medical Diagnosis model for COVID-19 infection detection based on Deep Features and Bayesian Optimization. *Appl. Soft Comput. J.* **2020**, *97*, 106580. [[CrossRef](#)]
18. Martínez, F.; Martínez, F.; Jacinto, E. Performance evaluation of the NASnet convolutional network in the automatic identification of COVID-19. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2020**, *10*, 662–667. [[CrossRef](#)]
19. Alazab, M.; Awajan, A.; Mesleh, A.; Abraham, A.; Jatana, V.; Alhyari, S. COVID-19 prediction and detection using deep learning. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2020**, *12*, 168–181.
20. Chowdhury, M.E.H.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Mahbub, Z.B.; Islam, K.R.; Khan, M.S.; Iqbal, A.; Emadi, N.A.; et al. Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access* **2020**, *8*, 132665–132676. [[CrossRef](#)]
21. Sethy, P.K.; Behera, S.K.; Ratha, P.K.; Biswas, P. Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. *Int. J. Math. Eng. Manag. Sci.* **2020**, *5*, 643–651. [[CrossRef](#)]
22. Toğaçar, M.; Ergen, B.; Cömert, Z. COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. *Comput. Biol. Med.* **2020**, *121*. [[CrossRef](#)] [[PubMed](#)]
23. Brunese, L.; Mercaldo, F.; Reginelli, A.; Santone, A. Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays. *Comput. Methods Programs Biomed.* **2020**, *196*, 105608. [[CrossRef](#)]
24. Minaee, S.; Kafieh, R.; Sonka, M.; Yazdani, S.; Jamalipour Soufi, G. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Med. Image Anal.* **2020**, *65*, 101794. [[CrossRef](#)]
25. Khan, A.I.; Shah, J.L.; Bhat, M.M. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Comput. Methods Programs Biomed.* **2020**, *196*, 105581. [[CrossRef](#)]

26. Pereira, R.M.; Bertolini, D.; Teixeira, L.O.; Silla, C.N.; Costa, Y.M.G. COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios. *Comput. Methods Programs Biomed.* **2020**, *194*, 105532. [[CrossRef](#)]
27. Ozturk, T.; Talo, M.; Yildirim, E.A.; Baloglu, U.B.; Yildirim, O.; Rajendra, A.U. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput. Biol. Med.* **2020**, *121*, 103792. [[CrossRef](#)]
28. Civit-Masot, J.; Luna-Perejón, F.; Morales, M.D.; Civit, A. Deep learning system for COVID-19 diagnosis aid using X-ray pulmonary images. *Appl. Sci.* **2020**, *10*, 4640. [[CrossRef](#)]
29. Oh, Y.; Park, S.; Ye, J.C. Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets. *IEEE Trans. Med. Imaging* **2020**, *39*, 2688–2700. [[CrossRef](#)]
30. Mohammed, M.A.; Abdulkareem, K.H.; Al-Waisy, A.S.; Mostafa, S.A.; Al-Fahdawi, S.; Dinar, A.M.; Alhakami, W.; Baz, A.; Al-Mhiqani, M.N.; Alhakami, H.; et al. Benchmarking Methodology for Selection of Optimal COVID-19 Diagnostic Model Based on Entropy and TOPSIS Methods. *IEEE Access* **2020**, *8*, 99115–99131. [[CrossRef](#)]
31. Rajaraman, S.; Siegelman, J.; Alderson, P.O.; Folio, L.S.; Folio, L.R.; Antani, S.K. Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-Rays. *IEEE Access* **2020**, *8*, 115041–115050. [[CrossRef](#)]
32. Loey, M.; Smarandache, F.; Eldeen, N.; Khalifa, M. A Deep Transfer Learning Model with Classical Data Augmentation and CGAN to Detect COVID-19 from Chest CT Radiography Digital Images. *Neural Comput. Appl.* **2020**, *2*, 1–17. [[CrossRef](#)] [[PubMed](#)]
33. Rahman, T.; Chowdhury, M.; Khandakar, A. *COVID-19 Radiography Database*; Kaggle: San Francisco, CA, USA, 2020.
34. Cohen, J.P.; Morrison, P.; Dao, L.; Roth, K.; Duong, T.Q.; Ghassemi, M. Covid-19 image data collection: Prospective predictions are the future. *arXiv* **2020**, arXiv:2006.11988.
35. Chung, A.; Wang, L.; Wong, A.; Lin, Z.Q.; McInnis, P.; Chung, A. *Actualmed COVID-19 Chest X-Ray Dataset Initiative*; GitHub: San Francisco, CA, USA, 2020.
36. Mooney, P. *Chest X-Ray Images (Pneumonia)*; Kaggle: San Francisco, CA, USA, 2018.
37. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Int. Conf. Mach. Learn.* **2015**, *37*, 448–456.
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
40. Esteban, C.; Hyland, S.L.; Rätsch, G. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv* **2017**, arXiv:1706.02633.
41. Sara, U.; Akter, M.; Uddin, M.S. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *J. Comput. Commun.* **2019**, *7*, 8–18. [[CrossRef](#)]
42. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
43. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Piscataway, NJ, USA, 21–26 July 2017; pp. 1800–1807.
44. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
45. Mahdianpari, M.; Salehi, B.; Rezaee, M.; Mohammadimanesh, F.; Zhang, Y. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sens.* **2018**, *10*, 1119. [[CrossRef](#)]
46. Gonzalez, T.F. Handb. Approx. Algorithms Metaheuristics. In *Handbook of Approximation Algorithms and Metaheuristics*; CRC Press: Boca Raton, FL, USA, 2007; pp. 1–1432.
47. Goutte, C.; Gaussier, É. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In Proceedings of the 27th European Conference on {IR} Research; {ECIR} 2005, Proceedings, Santiago de Compostela, Spain, 21–23 March 2005; Losada, D.E., Fernández-Luna, J.M., Eds.; Advances in Information Retrieval. Springer: Berlin/Heidelberg, Germany, 2005; Volume 3408, pp. 345–359.
48. Van Asch, V. *Macro-And Micro-Averaged Evaluation Measures*; CLiPS, University of Antwerp: Antwerp, Belgium, 2013; pp. 1–27.
49. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 1–48. [[CrossRef](#)]
50. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2020**, *17*, 168–192. [[CrossRef](#)]
51. Landis, J.R.; Koch, G.G. The Measurement of Observer Agreement for Categorical Data. *Biometrics* **1977**, *33*, 159. [[CrossRef](#)] [[PubMed](#)]
52. Zhang, S.; Cheng, D.; Jiang, D.; Kou, Q. Least Squares Relativistic Generative Adversarial Network for Perceptual Super-Resolution Imaging. *IEEE Access* **2020**, *8*, 185198–185208. [[CrossRef](#)]
53. Chen, X.; Duan, Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2180–2188.