Detection of COVID-19 Based on Synthetic Chest X-Ray (CXR) Images Using Deep Convolutional Generative Adversarial Networks (DCGAN) and Transfer Learning

Anhar and Dandi Septiandi
Department of Electrical Engineering, Universitas Riau, Jl. HR Soebrantas KM. 12.5, Pekanbaru 28293, Indonesia

ARTICLE INFO

Article history:

Received July 13, 2023
Revised August 17, 2023
Published September 09, 2023

Keywords:

COVID-19;
Detection;
Chest X-ray;
DCGAN;
Transfer learning

ABSTRACT

The global COVID-19 pandemic has significantly impacted the health and lives of people worldwide, with high numbers of cases and fatalities. Rapid and accurate diagnosis is crucially important. Radiographic imaging, particularly chest radiography (CXR), has been considered for diagnosing suspected COVID-19 patients. CXR images offers quick imaging, affordability, and wide accessibility, making it pivotal for screening. However, the scarcity of CXR images remains due to the pandemic's recent emergence. To address this scarcity, this study harnesses the capabilities of Deep Convolutional Generative Adversarial Networks (DCGAN). DCGAN is a convolution-based GAN approach, has the potential to alleviate the scarcity of CXR data by generating authentic-looking synthetic images. This study combines synthetic CXR images with real CXR images to bolster model performance, resulting in an Extended Dataset. Extended Dataset comprises 7,345 images, with 34.63% being original CXR images and 65.37% being synthetic images produced by DCGAN. Expanded Dataset then utilized to train three pre-trained models: ResNet50, EfficientNetV1, and EfficientNetV2. The outcomes are remarkable, showcasing considerable enhancement in detection accuracy. Especially for the EfficientNetV1 model, it takes the lead with an impressive accuracy of 99.21% after merely ten epochs, achieved within a brief training period of 6.18 minutes. This surpasses the prior accuracy of 98.43% observed when used the Original Dataset (without synthetic CXR images). Overall, this research offers a solution to mitigate the scarcity of synthetic CXR images for COVID-19 detection. For future endeavors, refining the quality of synthetic images stands as an area for exploration, enhancing the overall efficacy of this approach.

Corresponding Author:

Anhar, Universitas Riau, Jl. HR Soebrantas KM. 12.5, Pekanbaru 28293, Indonesia
Email: anhar@lecturer.unri.ac.id

1. INTRODUCTION

The COVID-19 pandemic has brought significant challenges to healthcare systems worldwide. With the rapid spread of the COVID-19 disease, fast and accurate diagnosis is crucial during this critical period [1].

One possible approach for COVID-19 detection is through analyzing radiographic images, specifically chest X-rays. Research by Fang [2] has shown that chest CT scans can be used to diagnose and assess COVID-19. Screening using chest CT scans has demonstrated higher sensitivity (98%) compared to Real-Time Reverse-Transcription Polymerase Chain Reaction (RT-PCR) tests [2]. Similarly, Xie [3] found that chest CT scans can indicate COVID-19 infection even when RT-PCR tests yield negative results.

However, with the increasing cases of COVID-19, routine CT usage burdens radiology departments and poses potential infection risks in the CT unit; thus, the need to identify COVID-19 infections in CXR images is growing [4]. CXR imaging has also been widely used for COVID-19 screening due to its shorter imaging time, lower cost, and widespread availability of X-ray scanners [5]. An automated detection method capable of identifying COVID-19 infections through CXR images is essential to curb the spread of COVID-19 cases and aid radiologists in patient diagnosis.

The application of Artificial Intelligence (AI) technology, particularly Deep Learning, has the potential to automate the COVID-19 diagnosis process. Deep learning has rapidly gained momentum in addressing various challenges in digital image and video analysis [6]. Specifically, it has been widely applied in tasks such as image classification [7], [8] and object detection [9].

Convolutional Neural Network (CNN) has emerged as a widely adopted technique in the field of Deep Learning [10], [11]. CNN and more recently transformers have garnered numerous achievements in the realm of image classification [12]. Several studies have explored the application of transfer learning. For instance, Aulia [13] proposes to develop a deep learning-based system for TB detection using X-ray images, achieving a remarkable accuracy of 99.76% in classifying normal and TB lungs with the VGG-16 architecture. Nandipati [14] aims to enhance the accuracy of lung cancer prognosis by employing the Accelerated Wrapper-based Binary Artificial Bee Colony algorithm (AWBABCA) for effective feature selection and the VGG19+CNN model for cancer classification. In another study, Alotaibi [15] demonstrated high accuracy (ranging from 97.23% to 98.3%) in detecting pneumonia, normal cases, and COVID-19 using four pre-trained models: ResNet50, VGG19, DenseNet121, and InceptionV3.

However, a significant challenge in COVID-19 detection using deep learning models is the limited availability and imbalance of medical image datasets, specifically CXR and CT images related to COVID-19 [16]. This was expected, the COVID-19 pandemic being a recent research topic, there is a limited availability of small, relevant, and properly labeled datasets, leading to a lack of standardized data [16]-[17]. Obtaining sufficient data for training deep learning models is costly and often restricted due to patient privacy concerns [17]. Furthermore, the analysis of COVID-19 Chest X-rays requires the expertise of specialized medical professionals, such as radiologists, who conduct meticulous manual examinations. As a result, the production of this medical data becomes both costly and time-consuming [18]. This challenge leads to imbalanced datasets, which can result in overfitting to the majority class. Insufficient data samples hinder meaningful learning in the clinical context [16].

Generative Adversarial Networks (GANs) have shown promise in addressing these challenges [16], [19]–[21]. GANs can generate synthetic images by learning from real data distribution [16]. The usage of Generative Adversarial Networks (GANs) in the context of medical images is still in its infancy, and the existing literature on this topic is relatively sparse. However, the potential of GANs in this domain is significant [22]. Several studies have developed GAN-based models using chest X-rays or CT scans as inputs. Waheed [19] developed an Auxiliary Classifier GAN (ACGAN) model called CovidGAN, which generated synthetic CXR images and improved the COVID-19 detection accuracy from 85% to 95%. Al-Shargabi [16] designed the COVID-CGAN model, which combined real and synthetic CXR images to achieve high detection accuracy using various deep-learning models. Kora Venu & Ravula [17] improved COVID-19 detection by augmenting CXR data using a generative modeling technique called Deep Convolutional Generative Adversarial Network (DCGAN). Puttagunta [20] utilized DCGAN and transfer learning on a dataset of chest X-ray images, improving the detection accuracy of COVID-19. Loey [23] also proposed a GAN-based model using a limited dataset and generated synthetic images, demonstrating its effectiveness in COVID-19 detection.

In recent studies, numerous techniques have been introduced in the literature using the emerging technologies of artificial intelligence to, detect, classify, and assess the diagnosis of COVID-19 from chest X-ray images. Table 1 provides an overview of studies pertaining to COVID-19, focusing on dataset size, the GAN architecture utilized to augment the dataset in GAN-involved cases, the classification technique adopted, and the performance of the classification model.

This paper proposes a comprehensive approach for improving COVID-19 detection using chest X-ray images. The primary research contributions of this study are as follows:

1. Expanding dataset that involves 4800 synthetic images and 2545 real CXR images and evaluation of Synthetic Images Impact; we investigate the impact of incorporating synthetic images on the overall model performance.
2. COVID-19 detection using a combination of real CXR images and synthetic CXR images generated by the DCGAN was demonstrated.

In this paper, a distinct methodology was adopted by merging synthetic and real CXR images to significantly enrich the dataset. The dataset encompasses 4800 synthetic CXR images and 2545 real CXR images. On the dataset, synthetic CXR images contributing to approximately 65% and real CXR images to about 35% of the total.

Table 1. A Summary of COVID-19-related study.

Journal Article	Dataset Size	GAN Used	Classification Method	Model Performance
[19]	1124 CXR images (403 COVID-CXR and 721 Normal-CXR) (82.9% train, 17,1% test)	COVID-GAN generated 1399 synthetic Normal-CXR and 1669 synthetic COVID-CXR	VGG-16	95% accuracy, 90% recall, 97% specificity
[16]	3290 CXR images (500 real images and 2790 generated images) (77.81% train, 22.19% test)	Conditional GAN (CGAN), generated 930 CXR synthetic COVID-19 images	InceptionResNetV2, Xception, SqueezeNet, VGG16, and AlexNet	99.72% accuracy, 99.73% precision, 99.73% recall
[20]	934 chest X-Ray images (234 COVID CXR and 700 Normal CXR) (70% train, 30% test)	DCGAN	VGG-16, InceptionV3 and MobilNet	Inception V3 95,90% accuracy
[23]	306 chest X-Ray images (69 COVID-19, 79 Pneumonia bacterial, 79 Pneumonia virus, 79 Normal) (90% train [divided by 80% train and 20% validation] and 10% test)	Proposed GAN, generated 7794 synthetic CXR images	Alexnet, Googlenet, and Restnet18	For 2 class (COVID-19 and Normal) Googlenet: 99.9% accuracy Alexnet: 99,6% accuracy Resnet18: 99.8% accuracy
[21]	742 CT images (345 COVID-19, 397 Non COVID-19) (57.28% train, 26.82% test, 15.90% validation)	CGAN, generated 2000 CT COVID-19 and 2000 CT Non Covid-19	AlexNet, VGGNet16, VGGNet19, GoogleNet, dan ResNet50	ResNet50: 77.39% accuracy, 61.70% recall, and 91.43% specificity
[24]	760 chest X-Ray images (280 COVID-19 CXR and 480 Normal CXR) (90% train, 20% test)	ACGAN, generated 500 synthetic COVID-19 CXR	VGG16, VGG19, ResNet50, Xception, InceptionV3	VGG19 99.38% accuracy, 99.38% precision, 99.38% recall, 99.38% F1-Score, 99.82% AUC
[25]	9544 chest X-Ray images (5500 Normal CXR and 4044 COVID-19 CXR) 80% training, 20% testing	CycleGAN	Inception V3, VGG16	Inception V3 94.2% accuracy, AUC = 92.2%
[26]	5910 chest X-Ray images (1143 COVID-19, 1345 Viral Pneumonia, 1341 Normal)	DCGAN	Proposed CNN, AlexNet and GoogLeNet	Proposed CNN: 98,5% accuracy, AlexNet: 96.7% accuracy and GoogLeNet: 95.5% accuracy
[27]	450 chest X-Ray images (150 COVID CXR, 150 Pneumonia and 150 Normal CXR) (70% train, 15% validation, 15% test)	GAN-based data augmentation, generated 560 synthetic CXR COVID-19, 560 synthetic CXR Pneumonia, and 560 synthetic CXR Normal	QuNet, DenseNet-121, ResNet101, InceptionV3, and Xception	QuNet 90% accuracy, 77.14% recall
[28]	423 chest X-Ray images (141 COVID-19, 141 Pneumonia, 141 Normal) (89.6% train, 10.4% test)	DCGAN	EfficientNet	96% AUC
Proposed research	2541 chest X-Ray images (1200 COVID-19, 1341 Normal) (80% train, 10% validation, 10% test)	DCGAN, generated 2400 synthetic COVID-19 CXR and 2400 synthetic Normal CXR	CNN, ResNet50, EfficientNet V1, dan EfficientNet V2	99.22% (*Accuracy*) 99.22 % (*Precision*) 99.21% (*Recall*) 99.21% (F1-*score*) 98.44% MCC

Moreover, the study significantly augments the scale of generated CXR images. In comparison to several previous studies, including Waheed utilization of COVID-GAN to create 1399 synthetic normal images and 1669 synthetic COVID-19 images [19], Al-Shargabi generating 930 synthetic COVID-19 images using CGAN

[16], and Karbhari produce 500 synthetic COVID-19 images using ACGAN [24]. This research demonstrates a substantial surge in the quantity of synthesized CXR images.

Diverging from prior methodologies that also utilizing DCGAN, this research uses ResNet50, EfficientNetV1, and EfficientNetV2. These models are separately harnessed, and their performances are meticulously compared to ascertain the superior classifier model. The model's performance is assessed using diverse metrics, encompassing accuracy, precision, recall, F1-score, ROC curve, and Matthews Correlation Coefficient (MCC).

This paper is an alternative approach to overcome the scarcity of medical image datasets using synthetic images generated by DCGAN and transfer learning. In brief, this paper will be structured as follows: Section 2 elaborates on the research methodology, including the research flow, overview dataset, data preprocessing, the process of synthesizing medical images using DCGAN, the development of a transfer learning-based detection model, model evaluation, and the deployment of the model. In Section 3, we delve into the outcomes of our synthetic CXR image generation using DCGAN. Additionally, we present the training and evaluation results obtained on both the original dataset and the extended dataset. The deployment of the detection model and subsequent discussions are also covered in this section. Finally, Section 4 encapsulates the paper with a concise conclusion.

2. METHODS

2.1. Research Flow

Fig. 1 illustrates the research workflow, which consists of several stages as follows. Firstly, the identification and clear definition of the problem to be addressed. Secondly, relevant datasets are collected. Thirdly, data pre-processing is performed through normalization, resizing, and classical data augmentation. Fourthly, the DCGAN model is utilized to generate synthetic data resembling the original dataset. Fifthly, the model is trained using transfer learning and CNN techniques. Sixthly, the best-performing model is saved. Finally, the model is implemented using the Streamlit framework.

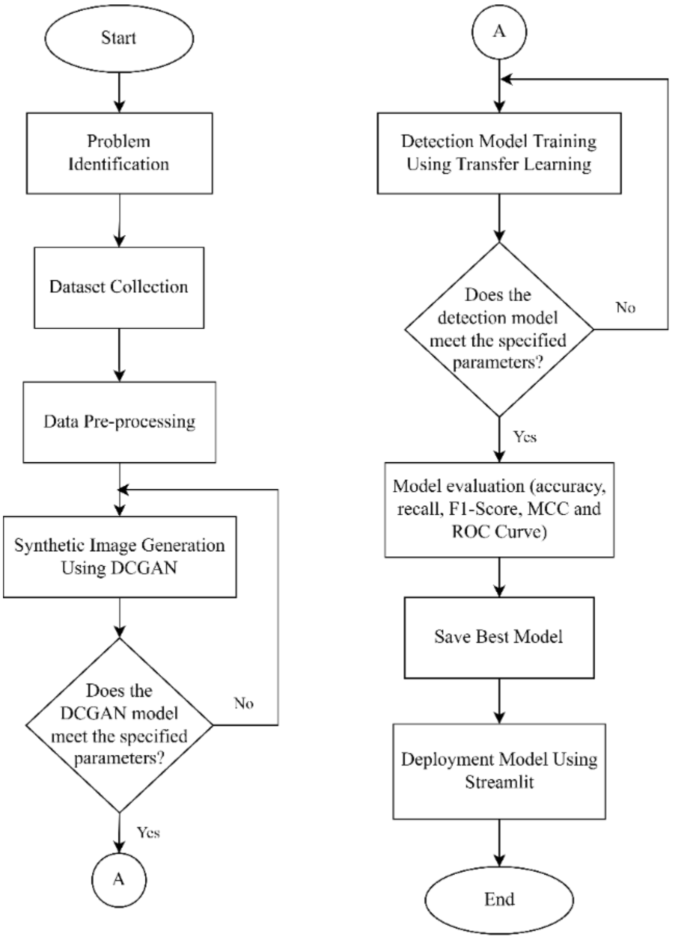

Start
Problem Identification
Dataset Collection
Data Pre-processing
Synthetic Image Generation Using DCGAN
Does the DCGAN model meet the specified parameters?
No
Yes
A
A
Detection Model Training Using Transfer Learning
Does the detection model meet the specified parameters?
No
Yes
Model evaluation (accuracy, recall, F1-Score, MCC and ROC Curve)
Save Best Model
Deployment Model Using Streamlit
End

Fig. 1. Research Flowchart

2.2. Dataset

This proposal utilizes the COVID-19 Radiography Database [28]-[29]. The sources of these datasets from the Italian Society of Medical and Interventional Radiology (SIRM) [29], the Novel Corona Virus 2019 dataset by Joseph Paul Cohen, Paul Morrison, and Lan Dao on GitHub [30], the Radiology Society of North America (RSNA) 2018 dataset [31], and the Kaggle chest X-Ray dataset [32].

COVID-19 Radiography Database consists of three classes: COVID-19 cases, Normal cases, and Viral Pneumonia cases. However, this study focuses on only two classes: COVID-19 and Normal cases. The dataset comprises approximately 2,545 images, with 1,200 CXR images for the COVID-19 class and 1,345 images for the Normal class.

In this research the dataset will be divided into three (train, validation and test). Initially, the model is trained using an appropriate training dataset specific to the given task. In the second phase, during the training process, the model is continuously evaluated on data that is not part of the training set, assessing its performance on previously unseen data. Finally, once the model has completed its training, it is put to the test using a test dataset, from which the final metrics are derived [33].

This research will consist of two datasets: the Original Dataset containing only original CXR data and the Extended Dataset combining original and synthetic CXR data. In the original dataset, the CXR images are divided into training data (960 images, 80%), validation data (120 images, 10%), and test data (120 images, 10%). The Expanded Dataset will include the addition of 4800 synthetic CXR images (2400 synthetic CXR COVID-19 and 2400 synthetic CXR Normal). These synthetic images will be distributed as 80% for training data and 20% for validation data from the total synthetic dataset. Thus, the training data will be a combination of 960 real CXR images and 1920 synthetic CXR images, while the validation data will include 120 real CXR images and 480 synthetic CXR images. It is important to note that the test data will solely consist of original images and will not include any synthetic images. This is aimed at objectively assessing the model's performance and testing its real-world applicability in identifying COVID-19 cases. By using only original images for the test data, we obtain a more authentic representation of the model's ability to handle challenges in real clinical settings. Table 2 presents the total data of real CXR images and synthetic CXR images for each dataset.

Table 2. Two type of dataset, Original Dataset (contain only real CXR image) and Extended Dataset (contain synthetic CXR images and real CXR image).

Dataset	Class	Train Data	Validation Data	Test Data	Percentage
Original Dataset	COVID-19 (1200 CXR)	960	120	120	100% original images, 0% synthetic images
	Normal (1345 CXR)	1072	135	135	
Extended Dataset	COVID-19 (1200 real CXR + 2400 synthetic CXR)	2880	600	120	34.63% original images, 65.37% synthetic images
	Normal (1345 real CXR + 2400 synthetic CXR)	2992	614	135	

2.3. Data Pre-processing

Preprocessing is a crucial step performed before entering the model training stages. During this process, digital images are processed to improve image quality and optimize the data for better results during the model training [34]. It included resizing all CXR images in the dataset to 224 × 224 pixels, matching the input requirements of pre-trained models. The images were then converted to float32 for improved numerical precision. Min-max normalization was applied to normalize the pixel values to a range of 0-1.

Overfitting is a primary challenge in deep learning. Overfitting is a condition in which the model's performance differs significantly between the training stage and the validation or test stages [35]. Classic data augmentation techniques can be employed to mitigate the effects of overfitting. The effective and efficiency of image augmentation have been substantiated [36]. The most common data augmentation techniques include scaling, cropping, flipping, and rotating [27]. In this research, two types of augmentation transformations are applied: RandomFlip, which randomly flips images both horizontally and vertically, and RandomZoom, a random zoom transformation that alters the image scale. The height_factor and width_factor parameters control the extent of vertical and horizontal enlargement or reduction of the images. These factors range from -0.1 to 0.2. By applying this augmentation layer to the training data before feeding it into the model, it generates a broader array of data variations. This process aids the model in learning more general patterns and enhances its overall ability to recognize new data effectively. This augmented transformations can be observed in Fig. 2.

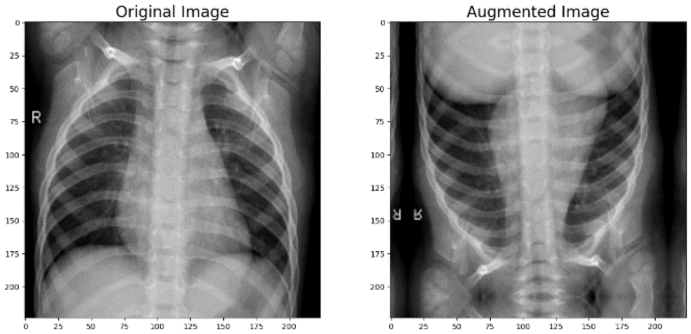


Fig. 2. Augmented image

2.4. Synthesis of Medical Images Using DCGAN

Deep Convolutional Generative Adversarial Network (DCGAN) is a convolution-based GAN approach proposed in 2015 by Radford [37]. The study by Radford [37] explores deep unsupervised CNN based on Generative Adversarial Networks (GAN). It aims to demonstrate the relevance of this architecture for unsupervised representation learning and address some training instabilities. Due to its strong training stability, DCGAN has been widely adopted in the medical imaging community [22]. Here is a detailed explanation of the DCGAN Generator architecture [37].

1. Linear Layer: The noise vector is fed into a fully connected layer, and the output is then reshaped into a 4D tensor.
2. Batch Normalization Layer: Stabilizes learning by normalizing inputs to have zero mean and unit variance, mitigating issues like vanishing or exploding gradients.
3. Up Sample Layer: A common technique for upsampling involves using transposed convolution. The purpose of this layer is to increase the size of the image from a smaller input.
4. 2D Convolutional Layer: This layer performs convolution operations on the 2D image. The Generator employs 2D convolutional layers to transform input noise into more realistic images.
5. ReLU Layer: The use of ReLU allows the model to saturate quickly and covers the color space of the training distribution.
6. TanH Activation: TanH activation function enables the model to achieve convergence more rapidly.

Layers 2 through 5 form the core generator block that can be repeated N times to obtain the desired output image shape. For the Discriminator architecture [37], the explanation is as follows:

1. Concat Layer: This layer combines both fake and real images in a single batch to feed into the discriminator. However, this can also be done separately to calculate the generator loss.
2. Convolutional Layer: Processes and extracts important features from images.
3. LeakyReLU: Allows gradient flow backward by introducing a small non-zero gradient. It helps mitigate the "dead neuron" problem where gradients become zero and learning stops for some neurons.
4. Dropout: Used during training to prevent overfitting. The model may memorize real image data, leading to training failures at that point as the discriminator can no longer be deceived by the generator.
5. Batch Normalization: Applied only at the end of each discriminator block except the first one. Applying batch normalization at every layer causes sample oscillations and model instability.
6. Linear Layer: A fully connected layer that takes the reshaped vector from the 2D batch normalization layer.
7. Sigmoid Activation: Utilized for the binary classification produced by the discriminator, hence the sigmoid layer.

Layers 2 through 5 form the core discriminator block, repeated N times to increase model complexity according to the training data [37].

The flowchart in Fig. 3 illustrates the step-by-step training process of the DCGAN in this study. It begins by generating random noise that serves as the initial input for the generator network. This noise is then fed into the generator network, which is then transformed into synthetic COVID-19/Normal chest X-ray (CXR) images. Simultaneously, both real CXR images and synthetic images produced by the generator are presented to the discriminator network. The primary role of the discriminator is to effectively differentiate between genuine and artificially generated images. Loss is computed to fine-tune the networks parameters through the process of backpropagation, and the discriminator is trained for 200 epochs. The model is saved after completion, enabling the generation of synthetic CXR images. The flowchart concludes the training process.

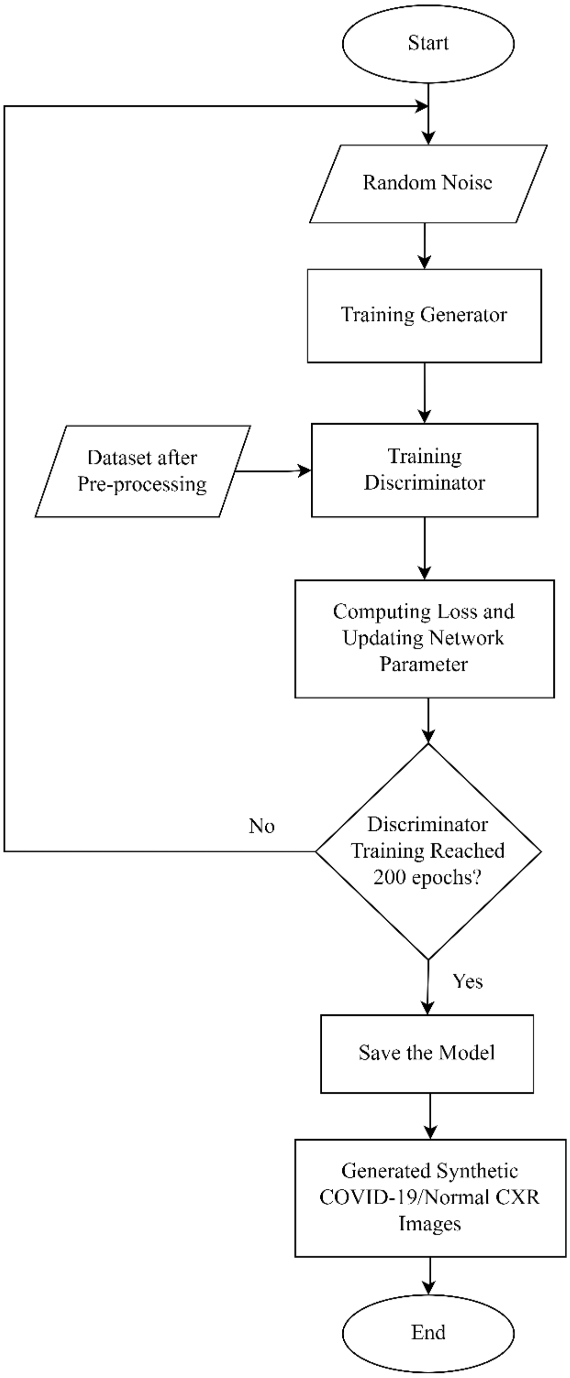
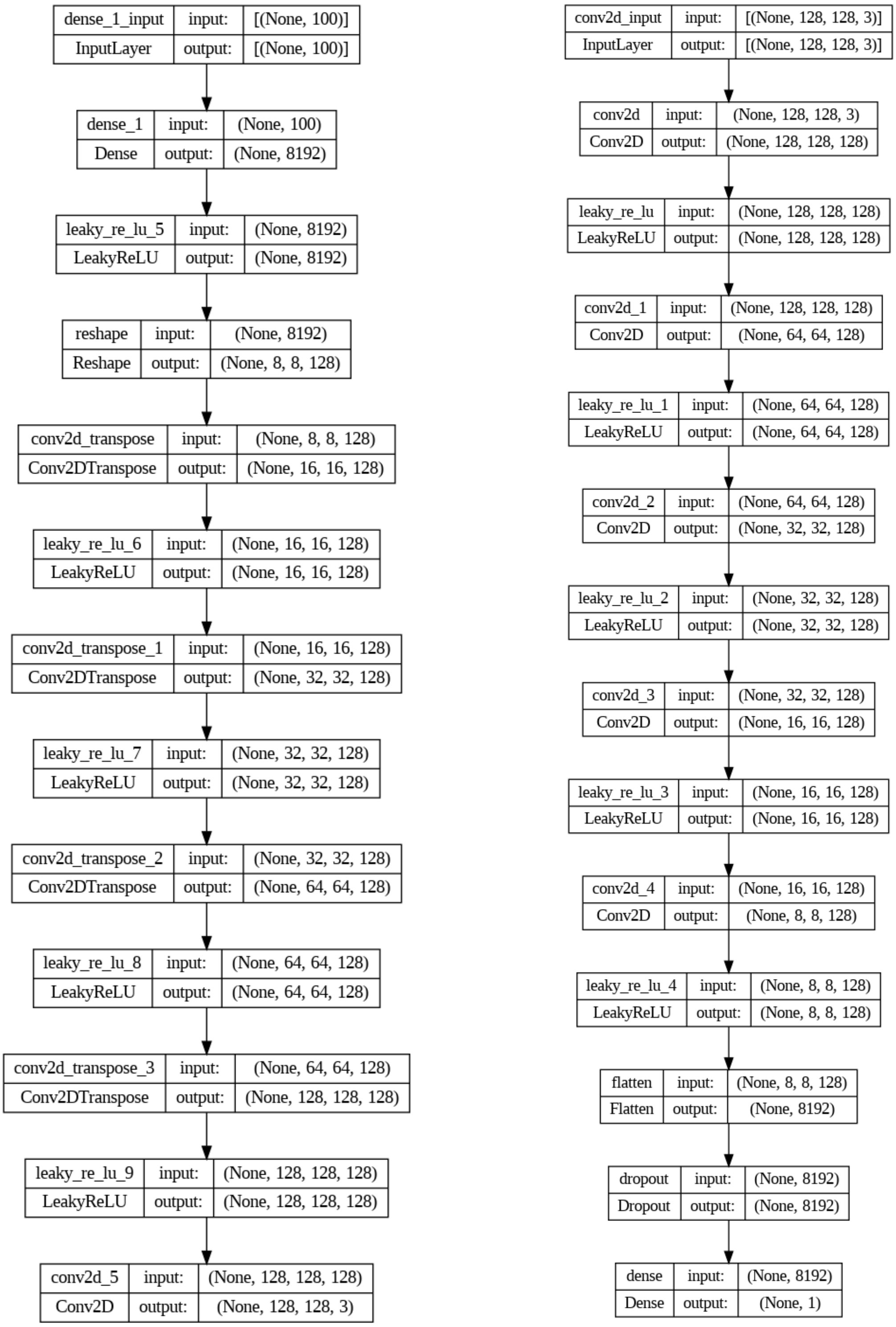

Start
Random Noise
Training Generator
Dataset after Pre-processing
Training Discriminator
Computing Loss and Updating Network Parameter
Discriminator Training Reached 200 epochs?
No
Yes
Save the Model
Generated Synthetic COVID-19/Normal CXR Images
End

Fig. 3. Training process of the DCGAN

In Fig. 4 show the architecture of DCGAN. DCGAN consists of two separate models: the Generator (G) and the Discriminator (D). The Generator aims to model a random noise vector as input and learn the data distribution to generate synthetic (fake) samples. On the other hand, the Discriminator takes both real training data and generated data (fake samples) and tries to classify them. The adversarial training process involves competition between these two models, where the loss of one model becomes the gain of the other model. This process drives the Generator to produce more realistic samples while the Discriminator becomes more adept at distinguishing between real and fake samples.


dense_1_input | input: | [(None, 100)]
InputLayer | output: | [(None, 100)]
dense_1 | input: | (None, 100)
Dense | output: | (None, 8192)
leaky_re_lu_5 | input: | (None, 8192)
LeakyReLU | output: | (None, 8192)
reshape | input: | (None, 8192)
Reshape | output: | (None, 8, 8, 128)
conv2d_transpose | input: | (None, 8, 8, 128)
Conv2DTranspose | output: | (None, 16, 16, 128)
leaky_re_lu_6 | input: | (None, 16, 16, 128)
LeakyReLU | output: | (None, 16, 16, 128)
conv2d_transpose_1 | input: | (None, 16, 16, 128)
Conv2DTranspose | output: | (None, 32, 32, 128)
leaky_re_lu_7 | input: | (None, 32, 32, 128)
LeakyReLU | output: | (None, 32, 32, 128)
conv2d_transpose_2 | input: | (None, 32, 32, 128)
Conv2DTranspose | output: | (None, 64, 64, 128)
leaky_re_lu_8 | input: | (None, 64, 64, 128)
LeakyReLU | output: | (None, 64, 64, 128)
conv2d_transpose_3 | input: | (None, 64, 64, 128)
Conv2DTranspose | output: | (None, 128, 128, 128)
leaky_re_lu_9 | input: | (None, 128, 128, 128)
LeakyReLU | output: | (None, 128, 128, 128)
conv2d_5 | input: | (None, 128, 128, 128)
Conv2D | output: | (None, 128, 128, 3)
conv2d_input | input: | [(None, 128, 128, 3)]
InputLayer | output: | [(None, 128, 128, 3)]
conv2d | input: | (None, 128, 128, 3)
Conv2D | output: | (None, 128, 128, 128)
leaky_re_lu | input: | (None, 128, 128, 128)
LeakyReLU | output: | (None, 128, 128, 128)
conv2d_1 | input: | (None, 128, 128, 128)
Conv2D | output: | (None, 64, 64, 128)
leaky_re_lu_1 | input: | (None, 64, 64, 128)
LeakyReLU | output: | (None, 64, 64, 128)
conv2d_2 | input: | (None, 64, 64, 128)
Conv2D | output: | (None, 32, 32, 128)
leaky_re_lu_2 | input: | (None, 32, 32, 128)
LeakyReLU | output: | (None, 32, 32, 128)
conv2d_3 | input: | (None, 32, 32, 128)
Conv2D | output: | (None, 16, 16, 128)
leaky_re_lu_3 | input: | (None, 16, 16, 128)
LeakyReLU | output: | (None, 16, 16, 128)
conv2d_4 | input: | (None, 16, 16, 128)
Conv2D | output: | (None, 8, 8, 128)
leaky_re_lu_4 | input: | (None, 8, 8, 128)
LeakyReLU | output: | (None, 8, 8, 128)
flatten | input: | (None, 8, 8, 128)
Flatten | output: | (None, 8192)
dropout | input: | (None, 8192)
Dropout | output: | (None, 8192)
dense | input: | (None, 8192)
Dense | output: | (None, 1)

(a) Generator (b) Discriminator

Fig. 4. DCGAN architecture

In this research, DCGAN utilized 2,545 images (1,200 COVID-19 CXR images and 1,345 normal CXR images) for training, with a distribution of 80% for training data, 10% for validation data, and 10% for testing data. The training process was conducted for 200 epochs, based on previous research by Rodriguez [38]. The utilized parameters for the DCGAN model can be observed in Table 3.

Table 3. DCGAN Configuration Parameters

Parameters	Value
Number of Epochs	200
Number of Channel	3
Batch Size	128
Optimizer	Adam Optimizer
Learning Rate of Generator	0.0002
Learning Rate of Discriminator	0.0002
Size of Latent Vector (z)	100

2.5. Transfer Learning-Based Detection Model

Transfer learning enables reusing a pre-trained model on a different but related task, often with minor adjustments. It proves especially valuable when the available training data is limited and frequently used in classifying medical images for rare or emerging diseases. Leveraging transfer learning requires only a few minor modifications to make the model perform better on the new task [39].

In detecting COVID-19 from chest X-ray (CXR) images, we employed transfer learning using three pre-trained models: ResNet50, EfficientNet V1, and EfficientNet V2.

2.5.1. ResNet50

ResNet is a deep convolutional neural network (CNN) introduced in 2015, developed by Kaiming He and his team. ResNet was designed to tackle the challenge of vanishing gradients in ultra-deep neural networks. ResNet achieved remarkable success, notably clinching the first position in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015 classification task, with a top-5 error rate as low as 3.57% (ensemble model). Furthermore, ResNet dominated the ILSVRC & COCO 2015 competitions, emerging as the champion in ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation tasks [40].

2.5.2. EfficientNetV1

EfficientNet V1, on the other hand, is a CNN designed for high efficiency and accuracy. It is based on the inverted bottleneck residual blocks from the MobileNetV2 model introduced in 2018. EfficientNetV1 employs a scaling method that uniformly adjusts all dimensions of depth, width, and resolution using compound coefficients [41]. The architecture of EfficientNet-B0 can be observed in Table 4.

Table 4. EfficientNet-B0 architecture.

Stage	Operator	Resolution	Channel	Layers
1	Conv3×3	224 × 224	32	1
2	MBConv1, k3×3	112 × 112	16	1
3	MBConv6, k3×3	112 × 112	24	2
4	MBConv6, k5×5	56 × 56	40	2
5	MBConv6, k3×3	28 × 28	80	3
6	MBConv6, k5×5	14 × 14	112	3
7	MBConv6, k5×5	14 × 14	192	4
8	MBConv6, k3×3	7 × 7	320	1
9	Conv1×1 & Pooling & FC	7 × 7	1280	1

EfficientNets have been systematically evaluated across eight widely adopted transfer learning datasets. Remarkably, the outcomes reveal that EfficientNets attain state-of-the-art accuracy in 5 out of the 8 datasets, including CIFAR-100 (91.7%) and Flowers (98.8%), all while maintaining a notably lean parameter count (reduced by up to 21 times). This substantiates the significant adaptability of EfficientNets for effective knowledge transfer [41]. In light of these findings, us expect that the implementation of EfficientNets promises to increase the level of accuracy in the data set under study.

2.5.3. EfficientNetV2

EfficientNet V2 is an efficient and accurate CNN based on the original EfficientNet introduced in 2019. In the architecture of EfficientNetV2, a non-uniform scaling strategy is employed to progressively add more layers. This approach differs from the EfficientNetV1 architecture, which simultaneously scales all stages using a simple compound scaling rule. The utilization of this method does not provide equal contributions to training speed and parameter efficiency. Table 5 showcases the architecture for EfficientNetV2, discovered using Neural Architecture Search (NAS), and introduces the novel Fused-MBConv operation [42].

Table 5. Architecture of EfficientNetV2.

Stage	Operator	Stride	Channels	Layers
0	Conv3×3	2	24	1
1	Fused-MBConv1, k3×3	1	24	2
2	Fused-MBConv4, k3×3	2	48	4
3	Fused-MBConv4, k3×3	2	64	4
4	MBConv4, k3×3, SE0.25	2	128	6
5	MBConv6, k3×3, SE0.25	1	160	9
6	MBConv6, k3×3, SE0.25	2	272	15
7	Conv1×1 & Pooling & FC	-	1792	1

All models (ResNet50, EfficientNetV1 and EfficientNetV2) were trained using the Adam optimizer, Binary Crossentropy loss, and sigmoid activation function. The model training was performed for ten epochs, and two types of callbacks were utilized in this study. The first callback used EarlyStopping, which monitors a specified validation metric (e.g validation loss or accuracy) during training and halts the training process if there is no improvement over a certain number of epochs. This prevents overfitting and saves computational resources by stopping training when the model's performance plateaus. The second callback is ReduceLROnPlateau to reduce the optimizer's learning rate if the evaluation metric did not improve for several epochs during training. ReduceLROnPlateau callback dynamically adjusts the learning rate to enhance convergence. These callbacks optimize training efficiency and model performance in COVID-19 detection. Table 6 provides a summary of the model configuration parameters to be used, while the flowchart illustrating the process of the detection model is presented in Fig. 5.

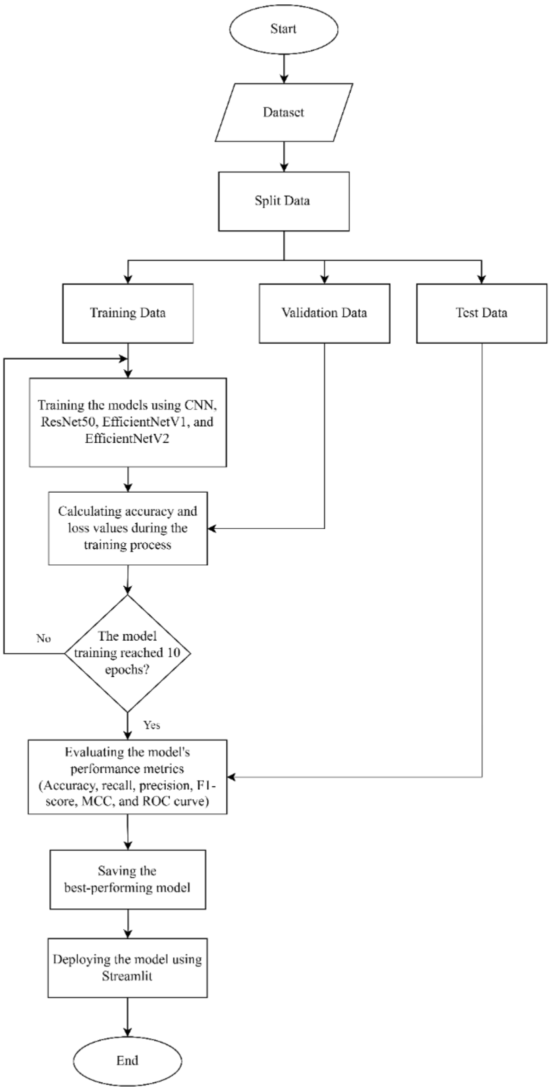

Start
Dataset
Split Data
Training Data
Validation Data
Test Data
Training the models using CNN, ResNet50, EfficientNetV1, and EfficientNetV2
Calculating accuracy and loss values during the training process
No
The model training reached 10 epochs?
Yes
Evaluating the model's performance metrics (Accuracy, recall, precision, F1-score, MCC, and ROC curve)
Saving the best-performing model
Deploying the model using Streamlit
End

Fig. 5. Process of the detection model

Table 6. Model Configuration Parameters.

Training Configuration	Values
Learning Rate	0.001
Epochs	10
Steps_per_epoch	131
Loss Function	Binary Crossentropy
Batch Size	32
Patience	2
Verbose	1
Callbacks	EarlyStopping and ReduceLROnPlateau
Optimizer	Adam optimizer

2.6. Model Evaluation

The application of evaluation metrics is crucial in gauging the classification performance of the trained model [33]. Each detection model will be evaluated by calculating several commonly used evaluation metrics in machine learning: accuracy, precision, recall, F1-score, ROC curve, and Matthews Correlation Coefficient (MCC). Out of the various evaluation metrics available, accuracy emerges as the predominant metric used for classification purposes [33]. However, it is recommended to implement alternative metrics to refine the assessment of classification performance [43].

Accuracy, a widely utilized metric in applications of Machine Learning (ML) in medicine, is defined as the ratio between the correctly classified samples and the total number of samples in the evaluation dataset (1). Precision represents the fraction of retrieved samples that are relevant and is computed as the ratio between correctly classified samples and all samples assigned to that specific class (2). Recall, also referred to as sensitivity or True Positive Rate (TPR), signifies the proportion of positive samples correctly classified. This metric holds significant importance in medical studies as it aims to minimize the omission of positive instances, resulting in a high recall rate (3). F1-score, a crucial metric, is the harmonic mean of precision and recall, offering a balanced evaluation that penalizes extreme values of either (4). Matthews Correlation Coefficient (MCC) is a comprehensive evaluation metric for binary and multiclass classification quality (5) [44].

The best values for accuracy, precision, recall, and F1-score are 1 (100%), while the worst values vary depending on the metric. Matthews Correlation Coefficient (MCC) has the best value of 1 (100%), while the worst value is -1 (-100%), indicating that the model is opposite to the actual values. The equations for accuracy, precision, recall, F1-score, and MCC are as follows [45], [46]:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1score = 2 \times \left(\frac{(Precision \times Recall)}{(Precision + Recall)}\right) \tag{4}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{5}$$

Were, true negatives (TN) represents the number of correctly classified CXR images of healthy patients (negative cases predicted as negative). At the same time, true positives (TP) indicates the number of correctly classified CXR images of COVID-19 patients (positive cases predicted as positive). False negatives (FN) represents the number of COVID-19 CXR images classified incorrectly (positive cases predicted as negative), and false positives (FP) indicates the number of healthy patient CXR images classified incorrectly (negative cases predicted as positive) [47]–[49].

The Receiver Operating Characteristic (ROC) curve is a graphical plot representing the model's ability to differentiate between positive and negative classes at various thresholds, with a best value of 1 (100%) and a worst value of 0.5, indicating random guessing performance. In the ROC curve, the X-axis represents the False Positive Rate (FPR) (6), and the Y-axis represents the True Positive Rate (TPR) (7) [39]. ROC curve is also known as recall or sensitivity. The curve is created by plotting TPR and FPR at various classification thresholds. The equations for TPR and FPR are as follows:

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

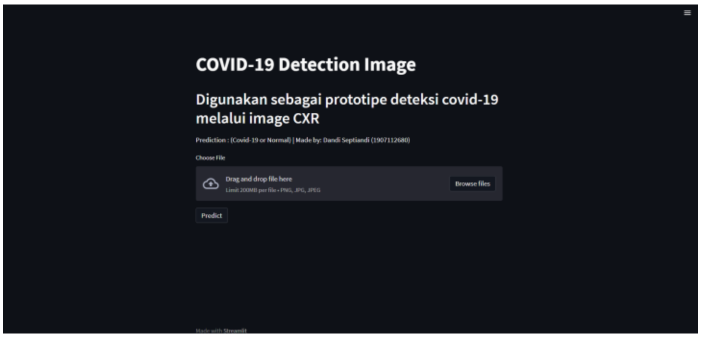
$$FPR = \frac{FP}{FP + TN} \tag{7}$$

2.7. Deployment Model

Streamlit is utilized to develop a web application that showcases the model and enables user interactions with the model. Streamlit-based web detection welcomes users with a user-friendly interface, allowing easy interaction with the application.

The interface of the COVID-19 detection system, deployed using Streamlit, is depicted in Fig. . Users have the option to upload CXR images for analysis by selecting a local image file. Once the CXR image is uploaded, the automated analysis will be conducted using a pre-trained model that has been trained before, and the prediction results are promptly displayed, showing the probabilities for the COVID-19 or Normal.

Deploying the model using Streamlit involves several steps, including model creation, using GitHub as a version control platform, and utilizing Streamlit to build a web application. The initial step is to create and train the model. The next step is to upload the source code and model files to a GitHub repository. This repository enables tracking of code and model changes over time and facilitates the deployment process. Once the GitHub repository is prepared, the next stage is to leverage Streamlit to build the web application.



COVID-19 Detection Image

Digunakan sebagai prototipe deteksi covid-19 melalui image CXR

Prediction : (Covid-19 or Normal) | Made by: Dandi Septiandi (1907112680)

Choose File

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

Fig. 6. User interface of COVID-19 Detection Image using Streamlit

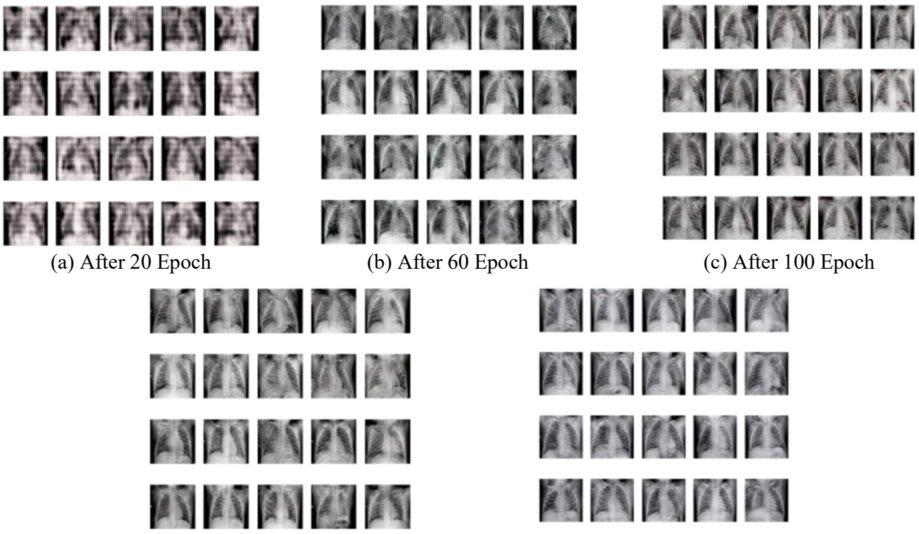
3. RESULTS AND DISCUSSION

3.1. Synthetic CXR Image Results Using DCGAN

In this study, DCGAN was trained for 200 epochs. During the early epochs, the Generator in DCGAN generated images with high noise, as seen in Fig. 7. It required several epochs for the Generator to start learning the basic structure of CXR images and produce synthetic CXR images that appear realistic and can deceive the Discriminator. It can be observed that the Generator continuously generates more realistic-looking images as the number of epochs increases. After the training is completed, the DCGAN model can generate synthetic CXR images. This study generated approximately 2400 synthetic CXR images for the COVID-19 class and 2400 synthetic CXR images for the Normal class.

During DCGAN training, the losses and accuracies of both the Discriminator and generator networks are simultaneously computed. Fig. illustrates the loss magnitudes during the training of the Generator and Discriminator. The graphs depict how the model learns and evolves throughout the training process. The generator loss graph indicates how well the Generator can produce synthetic images that resemble the original images. The main objective is minimizing generator loss, resulting in increasingly realistic synthetic images. On the other hand, the discriminator loss graph reflects the Discriminator's ability to differentiate between real and synthetic images. The Discriminator aims to maximize the discriminator loss by effectively distinguishing between real and synthetic images.

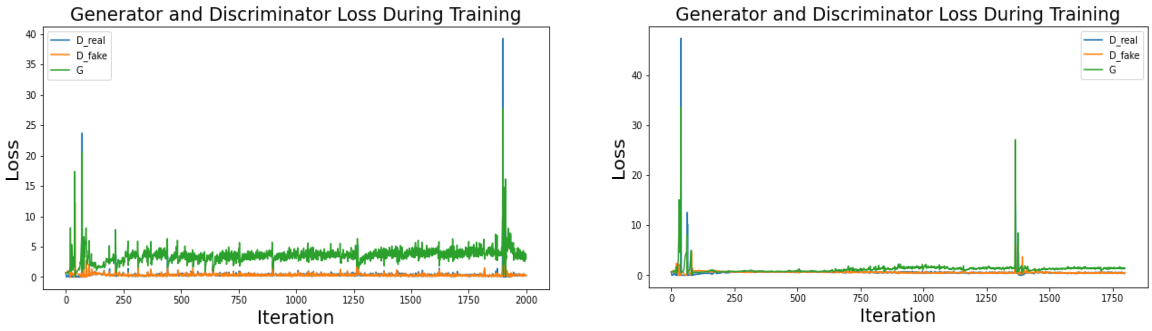
After the training, the DCGAN model can generate synthetic CXR images. This study will produce 2400 synthetic CXR images for COVID-19 and 2400 synthetic CXR images for the Normal class.



(a) After 20 Epoch (b) After 60 Epoch (c) After 100 Epoch

(d) After 160 Epoch (e) After 200 Epoch

Fig. 7. DCGAN learning process



(a) Normal (b) COVID-19

Fig. 8. The loss graph of the Generator and Discriminator in DCGAN

In the early iterations, as depicted in Fig. , the Generator may be unable to generate high-quality or realistic fake data that closely resembles the real data. As a result, the Discriminator can easily distinguish between fake and real data, leading to higher G and D_real losses compared to D_fake. This initial phase is crucial for the Generator to learn the distribution of real data and improve its ability to generate fake data that closely approximates it.

However, as the training progresses, the Generator gradually acquires better knowledge and skills in producing high-quality fake data. It is reflected in decreased G and D_real losses throughout the training. Although the G loss remains higher than D_real and D_fake losses, this can be attributed to the differences in the roles and complexities of each model. The Generator has a more complex task of emulating real data distribution, while the Discriminator only needs to discriminate between fake and real data.

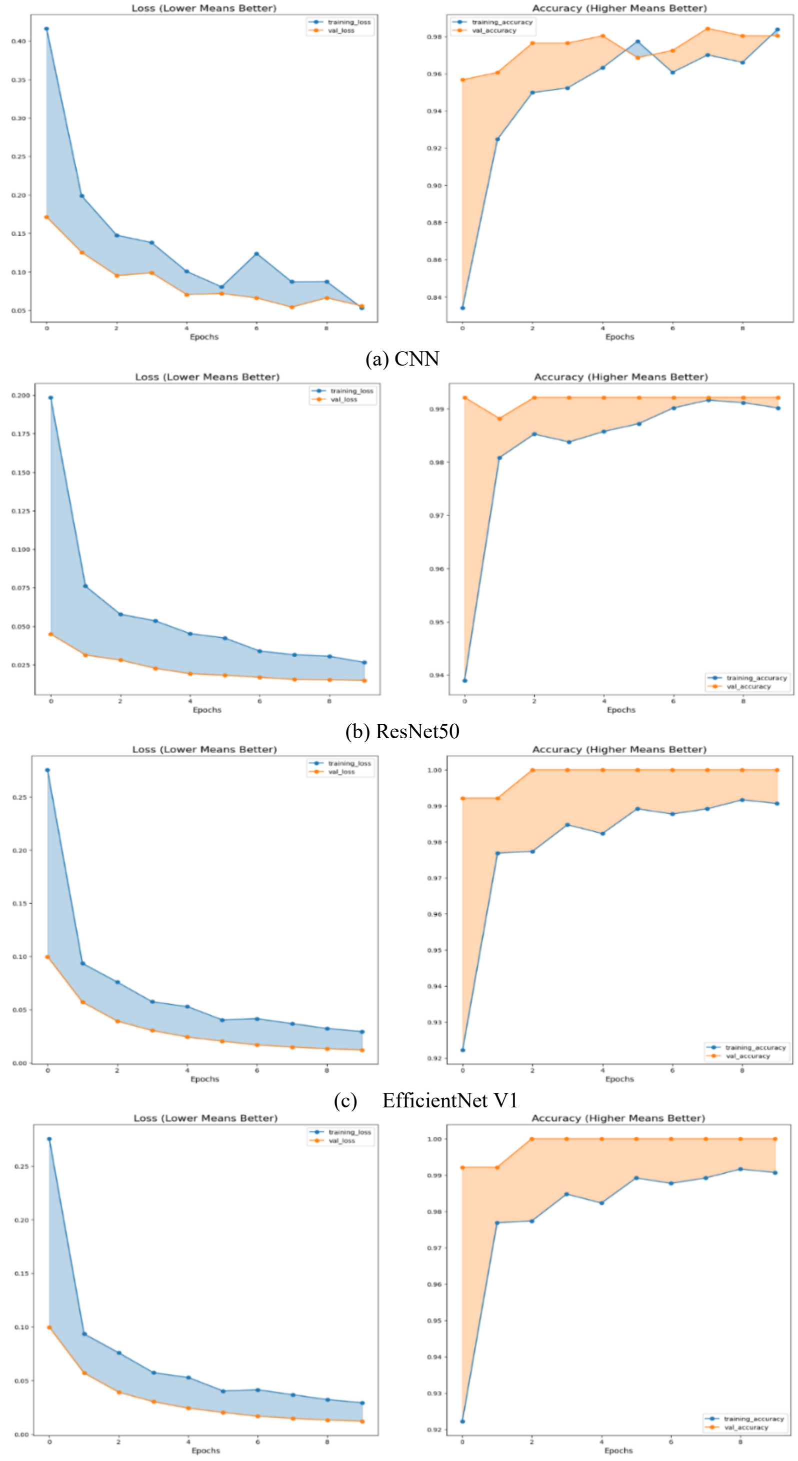
3.2. Training and Evaluation Results on the Original Dataset

The detection model was initially trained using the original dataset, which consisted of only real CXR images. The training process involved using 2,032 CXR images from the training data to learn patterns and features. The model was then evaluated during training using the validation data, which consisted of 255 CXR images.

The validation data played a crucial role in preventing overfitting during the training process. Overfitting occurs when the model becomes overly specialized in learning from the training data, resulting in predictions that fit the training data well but cannot be generalized to new data. The validation data was also used to evaluate the model's performance on unseen data. If the model's performance on the validation data starts to decline, it indicates the presence of overfitting. In such situations, various actions can be taken. The model can be retrained using different hyperparameters, or the training process can be stopped early to prevent further overfitting.

Fig. presents the performance graph of the model, showing the accuracy and loss values obtained when the model was evaluated using the validation data during the training process.

All models were trained using a batch size parameter of 32 and an epoch value of 10. The training results indicate no signs of overfitting for all models. It is demonstrated by the decreasing values of training_loss and val_loss as the number of epochs increases, while training_accuracy and val_accuracy show an improvement with increasing epochs. It indicates that the models are learning to generalize to new data rather than simply memorizing the training data. The training time for each model can be seen in Fig. 10.



(a) CNN

(b) ResNet50

(c) EfficientNet V1

(d) EfficientNet V2

Fig. 9. Accuracy and Loss Graphs During Training

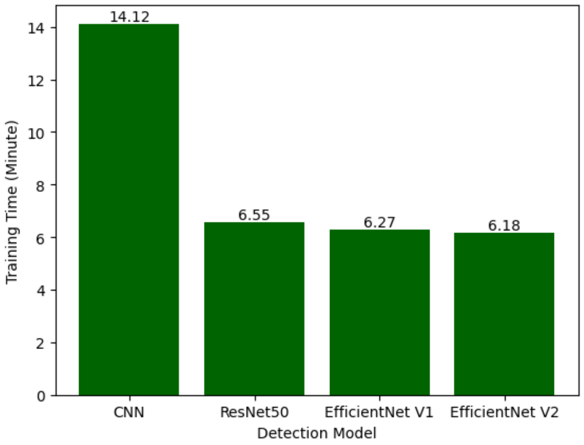
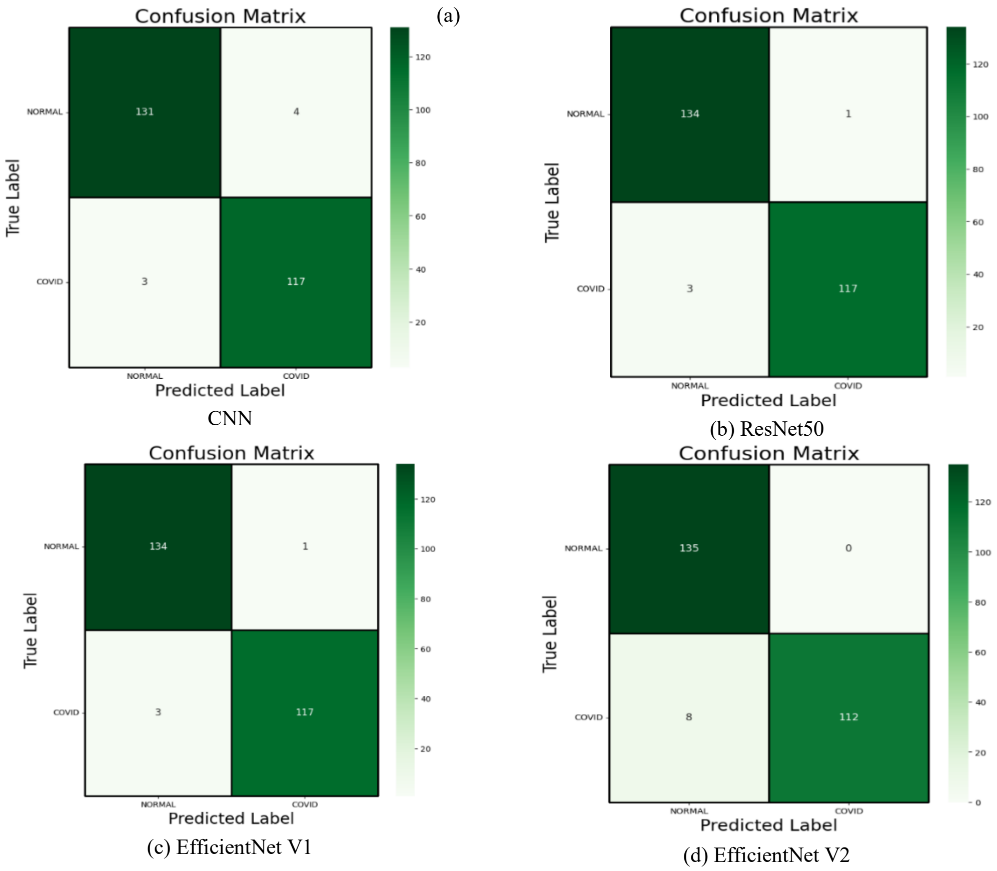


Fig. 10. The training time for each detection model using the original dataset

The trained detection model is then tested using a separate dataset called the test dataset, which consists of 255 CXR images (120 COVID-19 CXR and 135 Normal CXR). The test dataset is used to evaluate the final performance of the model. It is important to note that the test dataset was not used during training or to evaluate the model during training. We can obtain unbiased estimates of the model's performance on new data using a separate test dataset. The test set allows us to assess how well the model generalizes to unseen data and provides insights into its effectiveness in detecting COVID-19 cases and distinguishing them from normal cases. The confusion matrix of the four models is shown in Fig. 11.

Overall, as can be seen from Table 7, the performance of the four COVID-19 detection models on the original dataset shows that the ResNet50 V2 and EfficientNetV2 models outperform the other models with an accuracy of 98.43%, precision of 98.44%, recall of 98.43%, F1-score of 98.43%, and MCC of 96.86%.



(a)

CNN

(b) ResNet50

(c) EfficientNet V1

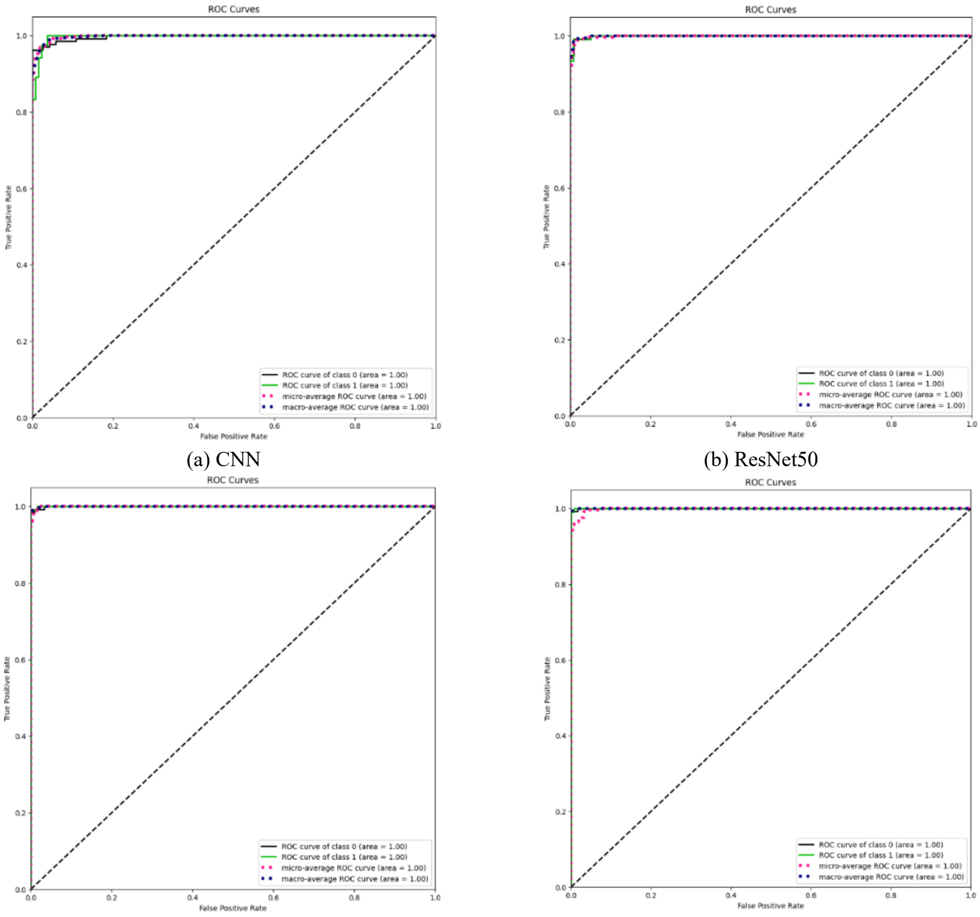
(d) EfficientNet V2

Fig. 11. Confusion Matrix of the detection model using the original dataset

Table 7. The results of testing the pre-trained model using the original dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC (%)
CNN	97.25	97.25	97.25	97.25	94.9
ResNet50	**98.43**	**98.44**	**98.43**	**98.43**	**96.86**
EfficientNetV1	**98.43**	**98.44**	**98.43**	**98.43**	**96.86**
EfficientNetV2	96.86	97.03	96.86	96.86	93.86

The ROC curve can be used to assess the overall comparison between the models at all possible threshold values. The ROC curves of the four models are shown in Fig. 12.



(a) CNN (b) ResNet50

(c) EfficientnetV1 (d) EfficientnetV2

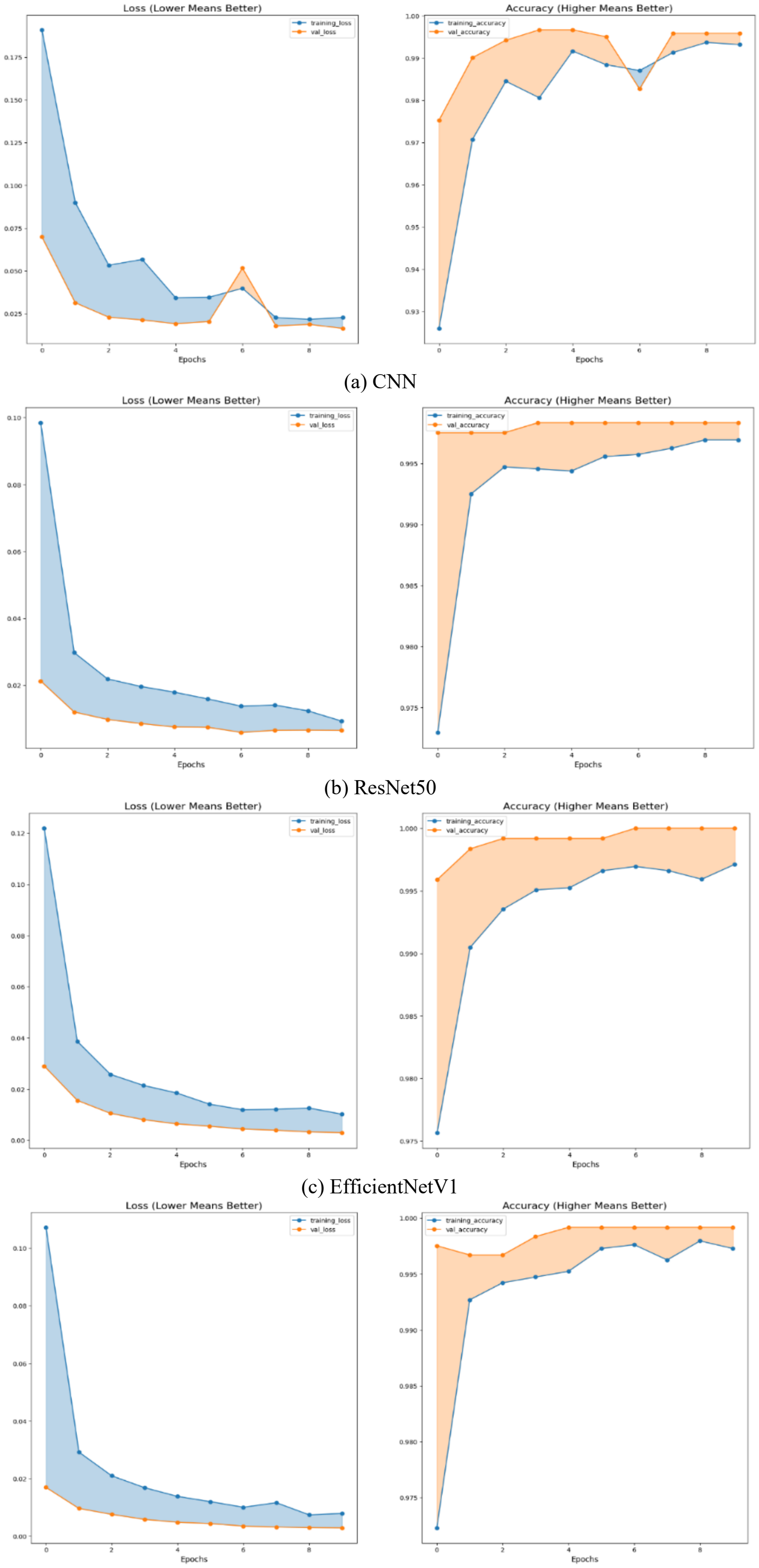
Fig. 12. The ROC curve of the detection model using the original dataset

In the ROC curve, the x-axis represents the False Positive Rate, the proportion of negatives incorrectly classified as positives. On the other hand, the y-axis represents the True Positive Rate, which is the proportion of positives that are correctly classified as positives. Ideally, a good model would have a high True Positive Rate and a low False Positive Rate. Therefore, the closer the curve is to the ROC curve's top-left corner (coordinate 0,1), the better the model's performance. Fig. 12 shows that all the models used in the study approach the top-left corner, indicating good model performance.

3.3. Training and Evaluation Results on the Extended Dataset

The following detection model is trained using the extended dataset, which combines original CXR images and synthetic CXR images. The impact of synthetic images to the pre-trained model is evaluated gradually by adding 2400 synthetic images. Each synthetic image is divided into 80% for training data and 20% for testing data and then merged with the previous original dataset.

The model is evaluated using validation data to assess accuracy and loss during the training process. The training will stop if the learning curve indicates signs of overfitting or underfitting. Fig. presents the performance graph of the model during training when using an extended dataset. The training time for each model can be seen in Fig. 14.



(a) CNN

(b) ResNet50

(c) EfficientNetV1

(d) EfficientNetV2

Fig. 13. Accuracy and Loss Graphs During Training using Extended Dataset

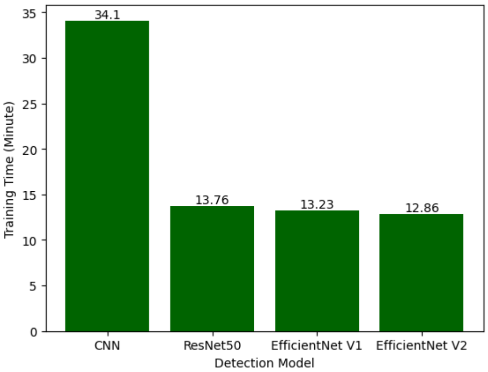


Fig. 14. The training time for each detection model using the extended dataset with the addition of 4800 synthetic images (2400 Covid-19 and 2400 Normal)

The shorter training time of the EfficientNet V2 model indicates better efficiency, while the longer training time of the CNN model may suggest higher complexity.

The trained detection models were further tested using a test data consisting of 255 CXR images (120 COVID-19 CXR and 135 Normal CXR). The test data was used to evaluate the final performance of the models. Table 8 presents the results of the testing for each model.

Table 8. Classification report for each model using the extended dataset.

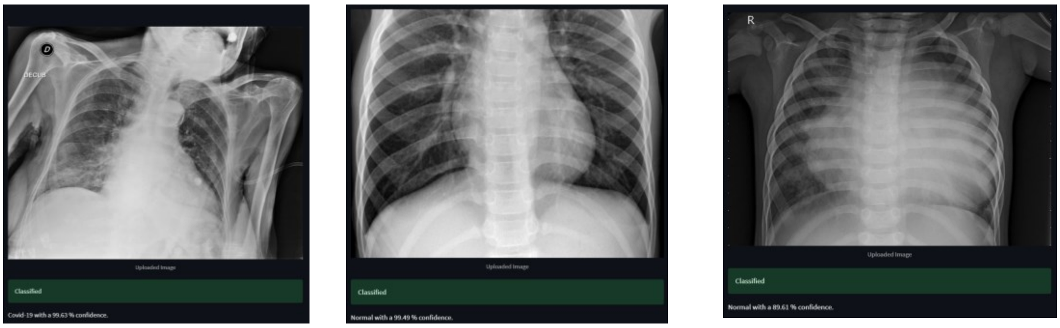
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC (%)
CNN	96.86	96.86	96.86	96.86	93.70
ResNet50	97.64	97.64	97.64	97.64	95.27
EfficientNetV1	**99.21**	**99.22**	**99.21**	**99.21**	**98.43**
EfficientNetV2	98.82	98.84	98.82	98.82	97.66

It's evident that the expanded dataset has led to varying degrees of improvement in model performance. Notably, EfficientNetV1 and EfficientNetV2 exhibit the most significant advancements, both exceeding 1% increase in all metrics. This emphasizes the effectiveness of the synthetic images generated by DCGAN, especially for these models, potentially leading to enhanced accuracy and robustness in medical image classification tasks. Further analysis and exploration of the specific augmentation techniques used can offer valuable insights for improving model performance in similar medical imaging contexts.

3.4. Deployment Detection Model

The deployment of the detection model is performed using the Streamlit application. The detection model can classify CXR images into COVID-19 or Normal classes.

The model is directly tested using three CXR images with different classes: COVID-19, Normal, and Pneumonia. The results show that the model can predict CXR images of COVID-19 with a similarity accuracy of 99.63% to the COVID-19 class. For Normal CXR images, the similarity accuracy to the Normal class is 99.49%. However, for Pneumonia CXR images, the model classifies them as Normal with similar accuracy to the Normal class of 89.61%. This study primary focus is on identifying COVID-19 cases, and any images not meeting the COVID-19 criteria will be considered as non-COVID-19 cases (normal). All the results of the model testing during the deployment process can be seen in Fig. 15.



Covid-19 with a 99.63 % confidence.

Normal with a 99.49 % confidence.

Normal with a 89.61 % confidence.

Fig. 15. The display of detection results using Streamlit

3.5. Discussion

In this study, we successfully generated synthetic images using DCGAN and transfer learning techniques for COVID-19 detection. The model achieved exceptional performance metrics when using extended dataset and EfficientNet V1, with an accuracy of 99.22%, precision of 99.22%, recall of 99.21%, and F1-score of 99.21%. These results indicate the robustness and reliability of our proposed method in accurately identifying COVID-19 cases based on chest X-ray images.

Our research outperformed several previous studies that utilized the same GAN method, DCGAN. Compared to Puttagunta's study [20], where an accuracy of 95.90% was achieved using the Inception V3 model on a dataset containing 934 CXR images, our model demonstrated significantly improved accuracy. Furthermore, the study by VJ & D [26] using the DCGAN method reported accuracies of only 98.5% for the CNN model, 96.7% for AlexNet, and 95.5% for GoogLeNet on a dataset containing 5910 CXR images, all of which were surpassed by our proposed approach.

Additionally, our research exhibited better results when compared to studies employing different GAN methods, such as the study by Waheed using the COVID-GAN method [19] which achieved 95% accuracy, 90% recall, and 97% specificity. Additionally, this study outperforms the study by Loey using the CGAN method [21] with 77.39% accuracy, 61.70% recall, and 91.43% specificity.

Furthermore, this research outperforms several previous studies in terms of model training time. This study only requires approximately 13.23 minutes for model training using EfficientNet V1. In comparison, the study by Al-Shargabi [16] took about 143 minutes for InceptionResNetV2 training, VJ & D [26] required approximately 1960 minutes for CNN, 2563 minutes for AlexNet, and 2660 minutes for GoogLeNet training. Additionally, the study by Gulakala [50] took 210 minutes for training using the proposed CNN.

The remarkable performance of our proposed method has important implications for COVID-19 detection systems. The integration of DCGAN-generated synthetic images with EfficientNet V1 enhances the model's accuracy, even in cases of limited real medical image datasets. This advancement holds significant potential to streamline the diagnostic process and aid healthcare professionals in making more accurate and timely COVID-19 diagnoses.

Our research possesses several strengths, including the successful utilization of DCGAN-generated synthetic images, which mitigates the challenges posed by limited real datasets. Additionally, the reduced model training time with EfficientNet V1 demonstrates the efficiency of our approach. However, it is essential to acknowledge some limitations, such as the necessity for larger and more diverse real CXR images datasets to further improve generalization and the quality of the generated synthetic CXR images needs to be improved using different approaches.

Overall, our study's main findings reveal the effectiveness of our proposed method for COVID-19 detection with limited chest X-ray images. Despite certain limitations, our research contributes valuable insights for future advancements, paving the way for more accurate and efficient COVID-19 diagnostic tools.

4. CONCLUSION

This study successfully addresses the critical challenge of limited availability of CXR data for COVID-19 detection by leveraging DCGAN to generate a more extensive dataset. The Extended Dataset, consisting of 2400 synthetic CXR images for each category (COVID-19 and Normal), significantly enhances the training process and performance evaluation. Through the evaluation of transfer learning methods, including ResNet50 V2, EfficientNet V1, and EfficientNet V2, we have identified the EfficientNet V1 model as the most effective, achieving exceptional performance metrics on the Extended Dataset. Notably, the model achieved an accuracy, precision, recall, F1-score, and MCC of 99.21%, 99.22%, 99.21%, 99.21%, and 98.43%, respectively. These findings demonstrate the immense potential of synthetic image generation and underscore the effectiveness of deep learning models in accurately detecting COVID-19.

Future work in this domain can be directed towards further improving the synthetic image generation process using advanced GAN techniques and exploring other transfer learning models to enhance detection accuracy. Moreover, conducting a thorough analysis of the model's generalizability across diverse datasets and clinical settings would contribute to the practical applicability of the proposed method.

Despite the significant contributions of this study, some limitations should be acknowledged. Firstly, the reliance on pre-existing GAN methods might restrict the model's flexibility in handling diverse datasets. Further research could focus on developing novel GAN architectures tailored to medical imaging tasks. Additionally, the synthetic images' quality and diversity can be further optimized to ensure better generalization and real-world applicability.

In summary, this research presents an approach to overcome the scarcity of medical image datasets using DCGAN-generated synthetic images and transfer learning. The results highlight the model's outstanding performance in detecting COVID-19 cases accurately. By providing a larger and more diverse dataset, this study contributes in the domain of automated COVID-19 detection. The successful integration of synthetic CXR images by DCGAN and transfer learning showcases the potential for data augmentation techniques in medical image analysis, fostering further research in this vital area. These findings hold substantial implications for the development of efficient and reliable diagnostic systems to combat the ongoing pandemic and other infectious diseases in the future.

REFERENCES

[1] S. Tong, C. Amand, A. Kieffer, and M. H. Kyaw, "Trends in healthcare utilization and costs associated with pneumonia in the United States during 2008–2014," *BMC Health Serv. Res.*, vol. 18, no. 1, p. 715, 2018, https://doi.org/10.1186/s12913-018-3529-4.

[2] Y. Fang *et al.*, "Sensitivity of Chest CT for COVID-19: Comparison to RT-PCR," *Radiology*, vol. 296, no. 2, pp. E115–E117, 2020, https://doi.org/10.1148/radiol.2020200432.

[3] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, and J. Liu, "Chest CT for Typical Coronavirus Disease 2019 (COVID-19) Pneumonia: Relationship to Negative RT-PCR Testing," *Radiology*, vol. 296, no. 2, pp. E41–E45, 2020, https://doi.org/10.1148/radiol.2020200343.

[4] Y. Oh, S. Park, and J. C. Ye, "Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets," *IEEE Trans. Med. Imaging*, vol. 39, no. 8, pp. 2688–2700, 2020, https://doi.org/10.1109/TMI.2020.2993291.

[5] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, and R. B. Pachori, "Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study," *Biomed. Signal Process. Control*, vol. 64, p. 102365, 2021, https://doi.org/10.1016/j.bspc.2020.102365.

[6] S. López-Tapia, R. Molina, and A. K. Katsaggelos, "Deep learning approaches to inverse problems in imaging: Past, present and future," *Digit. Signal Process.*, vol. 119, p. 103285, 2021, https://doi.org/10.1016/j.dsp.2021.103285.

[7] H. I. K. Fathurrahman, A. Ma'arif, and L.-Y. Chin, "The Development of Real-Time Mobile Garbage Detection Using Deep Learning," *J. Ilm. Tek. Elektro Komput. Dan Inform.*, vol. 7, no. 3, p. 472, 2022, https://doi.org/10.26555/jiteki.v7i3.22295.

[8] R. Syah and A.-K. Al-Khowarizmi, "Optimization of Applied Detection Rate in the Simple Evolving Connectionist System Method for Classification of Images Containing Protein," *J. Ilm. Tek. Elektro Komput. Dan Inform.*, vol. 7, no. 1, p. 154, 2021, https://doi.org/10.26555/jiteki.v7i1.20508.

[9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018, https://doi.org/10.48550/ARXIV.1804.02767.

[10] M. Z. Alom *et al.*, "A State-of-the-Art Survey on Deep Learning Theory and Architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019, https://doi.org/10.3390/electronics8030292.

[11] T. M. Saravanan, K. Karthiha, R. Kavinkumar, S. Gokul, and J. P. Mishra, "A novel machine learning scheme for face mask detection using pretrained convolutional neural network," *Mater. Today Proc.*, vol. 58, pp. 150–156, 2022, https://doi.org/10.1016/j.matpr.2022.01.165.

[12] J. Plested and T. Gedeon, "Deep transfer learning for image classification: a survey," *arXiv preprint arXiv:2205.09904*, 2022, https://doi.org/10.48550/ARXIV.2205.09904.

[13] S. Aulia and S. Hadiyoso, "Tuberculosis Detection in X-Ray Image Using Deep Learning Approach with VGG-16 Architecture," *J. Ilm. Tek. Elektro Komput. Dan Inform.*, vol. 8, no. 2, p. 290, 2022, https://doi.org/10.26555/jiteki.v8i2.23994.

[14] B. L. Nandipati and N. Devarakonda, "VGG19+CNN: Deep Learning-Based Lung Cancer Classification with Meta-Heuristic Feature Selection Methodology," *Indones. J. Electr. Eng. Inform. IJEEI*, vol. 11, no. 1, pp. 248–266, 2023, https://doi.org/10.52549/ijeei.v11i1.4394.

[15] A. Alotaibi, "Transfer Learning for Detecting Covid-19 Cases Using Chest X-Ray Images," *Int. J. Mach. Learn. Networked Collab. Eng.*, vol. 4, no. 1, pp. 21–29, 2020, https://doi.org/10.30991/IJMLNCE.2020v04i01.003.

[16] A. A. Al-Shargabi, J. F. Alshobaili, A. Alabdulatif, and N. Alrobah, "COVID-CGAN: Efficient Deep Learning Approach for COVID-19 Detection Based on CXR Images Using Conditional GANs," *Appl. Sci.*, vol. 11, no. 16, p. 7174, 2021, https://doi.org/10.3390/app11167174.

[17] S. Kora Venu and S. Ravula, "Evaluation of Deep Convolutional Generative Adversarial Networks for Data Augmentation of Chest X-ray Images," *Future Internet*, vol. 13, no. 1, p. 8, 2020, https://doi.org/10.3390/fi13010008.

[18] M. F. Ng and C. A. Hargreaves, "Generative Adversarial Networks for the Synthesis of Chest X-ray Images," in *The 3rd International Electronic Conference on Applied Sciences*, p. 84, 2023, https://doi.org/10.3390/ASEC2022-13954.

[19] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection," *IEEE Access*, vol. 8, pp. 91916–91923, 2020, https://doi.org/10.1109/ACCESS.2020.2994762.

[20] M. Puttagunta, R. Subban, and N. K. B. C, "A Novel COVID-19 Detection Model Based on DCGAN and Deep Transfer Learning," *Procedia Comput. Sci.*, vol. 204, pp. 65–72, 2022, https://doi.org/10.1016/j.procs.2022.08.008.

[21] M. Loey, G. Manogaran, and N. E. M. Khalifa, "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, https://doi.org/10.1007/s00521-020-05437-x.

[22] X. Yi, E. Walia, and P. Babyn, "Generative Adversarial Network in Medical Imaging: A Review," *Med. Image Anal.*, vol. 58, p. 101552, 2019, https://doi.org/10.1016/j.media.2019.101552.

[23] M. Loey, F. Smarandache, and N. E. M. Khalifa, "Within the Lack of Chest COVID-19 X-ray Dataset: A Novel Detection Model Based on GAN and Deep Transfer Learning," *Symmetry*, vol. 12, no. 4, p. 651, 2020, https://doi.org/10.3390/sym12040651.

[24] Y. Karbhari, A. Basu, Z. W. Geem, G.-T. Han, and R. Sarkar, "Generation of Synthetic Chest X-ray Images and Detection of COVID-19: A Deep Learning Based Approach," *Diagnostics*, vol. 11, no. 5, p. 895, 2021, https://doi.org/10.3390/diagnostics11050895.

[25] G. Bargshady, X. Zhou, P. D. Barua, R. Gururajan, Y. Li, and U. R. Acharya, "Application of CycleGAN and Transfer Learning Techniques for Automated Detection of COVID-19 using X-ray Images," *Pattern Recognit. Lett.*, vol. 153, pp. 67–74, 2022, https://doi.org/10.1016/j.patrec.2021.11.020.

[26] S. VJ and J. F. D, "Deep Learning Algorithm for COVID-19 Classification Using Chest X-Ray Images," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–10, 2021, https://doi.org/10.1155/2021/9269173.

[27] U. Asghar, M. Arif, K. Ejaz, D. Vicoveanu, D. Izdrui, and O. Geman, "An Improved COVID-19 Detection using GAN-Based Data Augmentation and Novel QuNet-Based Classification," *BioMed Res. Int.*, vol. 2022, pp. 1–9, 2022, https://doi.org/10.1155/2022/8925930.

[28] P. M. Shah *et al.*, "DC-GAN-based synthetic X-ray images augmentation for increasing the performance of EfficientNet for COVID-19 detection," *Expert Syst.*, vol. 39, no. 3, 2022, https://doi.org/10.1111/exsy.12823.

[29] R. Chen *et al.*, "Risk Factors of Fatal Outcome in Hospitalized Subjects With Coronavirus Disease 2019 From a Nationwide Analysis in China," *Chest*, vol. 158, no. 1, pp. 97–105, 2020, https://doi.org/10.1016/j.chest.2020.04.010.

[30] J. C. Monteral, "COVID-Chestxray Database," 2020. [Online]. Available: https://github.com/ieee8023/covid-chestxray-dataset.

[31] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2097–2106, 2017, https://doi.org/10.48550/ARXIV.1705.02315.

[32] P. Mooney, "Chest X-Ray Images (Pneumonia)," 2018. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[33] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci. Rep.*, vol. 12, no. 1, p. 5979, 2022, https://doi.org/10.1038/s41598-022-09954-8.

[34] T. Singh, K. Kumar, and S. Bedi, "A Review on Artificial Intelligence Techniques for Disease Recognition in Plants," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, p. 012032, 2021, https://doi.org/10.1088/1757-899X/1022/1/012032.

[35] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J. Imaging*, vol. 9, no. 2, p. 46, 2023, https://doi.org/10.3390/jimaging9020046.

[36] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, p. 109347, 2023, https://doi.org/10.1016/j.patcog.2023.109347.

[37] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv preprint arXiv:1511.06434*, 2015, https://doi.org/10.48550/ARXIV.1511.06434.

[38] J.-A. Rodrıguez-de-la-Cruz, H.-G. Acosta-Mesa, E. Mezura-Montes, F. A. Cosıo, B. Escalante-Ramırez, and J. O. Montiel, "Evolution of conditional-DCGANs for the synthesis of chest X-ray images.," *Proc. SPIE*, 2021, https://doi.org/10.1117/12.2606272.

[39] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. Jamalipour Soufi, "Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning," *Med. Image Anal.*, vol. 65, p. 101794, 2020, https://doi.org/10.1016/j.media.2020.101794.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016, https://doi.org/10.1109/CVPR.2016.90.

[41] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Int. Conf. Mach. Learn.*, pp. 6105–6114, 2019, https://doi.org/10.48550/ARXIV.1905.11946.

[42] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," *Int. Conf. Mach. Learn.*, pp. 10096-10106, 2021, https://doi.org/10.48550/ARXIV.2104.00298.

[43] C. Bi, S. Xu, N. Hu, S. Zhang, Z. Zhu, and H. Yu, "Identification Method of Corn Leaf Disease Based on Improved Mobilenetv3 Model," *Agronomy*, vol. 13, no. 2, p. 300, 2023, https://doi.org/10.3390/agronomy13020300.

[44] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment," *IEEE Access*, vol. 9, pp. 78368–78381, 2021, https://doi.org/10.1109/ACCESS.2021.3084050.

[45] A. S. Musallam, A. S. Sherif, and M. K. Hussein, "Efficient framework for detecting COVID-19 and pneumonia from chest X-ray using deep convolutional network," *Egypt. Inform. J.*, vol. 23, no. 2, pp. 247–257, 2022, https://doi.org/10.1016/j.eij.2022.01.002.

[46] D. K. Atal and M. Singh, "Arrhythmia Classification with ECG signals based on the Optimization-Enabled Deep Convolutional Neural Network," *Comput. Methods Programs Biomed.*, vol. 196, p. 105607, 2020, https://doi.org/10.1016/j.cmpb.2020.105607.

[47] E. F. Ohata *et al.*, "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning," *IEEECAA J. Autom. Sin.*, vol. 8, no. 1, pp. 239–248, 2021, https://doi.org/10.1109/JAS.2020.1003393.

[48] A. Elazab, M. A. Elfattah, and Y. Zhang, "Novel multi-site graph convolutional network with supervision mechanism for COVID-19 diagnosis from X-ray radiographs," *Appl. Soft Comput.*, vol. 114, p. 108041, 2022, https://doi.org/10.1016/j.asoc.2021.108041.

[49] S. Showkat and S. Qureshi, "Efficacy of Transfer Learning-based ResNet models in Chest X-ray image classification for detecting COVID-19 Pneumonia," *Chemom. Intell. Lab. Syst.*, vol. 224, p. 104534, 2022, https://doi.org/10.1016/j.chemolab.2022.104534.

[50] R. Gulakala, B. Markert, and M. Stoffel, "Generative Adversarial Network Based Data Augmentation for CNN Based Detection of COVID-19," *Sci. Rep.*, vol. 12, no. 1, p. 19186, 2022, https://doi.org/10.1038/s41598-022-23692-x.