Realistic image synthesis of COVID-19 chest X-rays using depthwise boundary equilibrium generative adversarial networks

Zendi Iklima, Trie Maya Kadarina, Eko Ihsanto
Department of Electrical Engineering, Faculty of Engineering, Universitas Mercu Buana, Jakarta, Indonesia

Article Info

Article history:

Received May 30, 2021
Revised Jun 8, 2022
Accepted Jun 25, 2022

Keywords:

Chest X-ray
Convolutional neural network
COVID-19 virus
DepthwiseBEGAN
Image synthesis

ABSTRACT

Researchers in various related fields research preventing and controlling the spread of the coronavirus disease (COVID-19) virus. The spread of the COVID-19 is increasing exponentially and infecting humans massively. Preliminary detection can be observed by looking at abnormal conditions in the airways, thus allowing the entry of the virus into the patient's respiratory tract, which can be represented using computer tomography (CT) scan and chest X-ray (CXR) imaging. Particular deep learning approaches have been developed to classify COVID-19 CT or CXR images such as convolutional neural network (CNN), and deep convolutional neural network (DCNN). However, COVID-19 CXR dataset was measly opened and accessed. Particular deep learning method performance can be improved by augmenting the dataset amount. Therefore, the COVID-19 CXR dataset was possibly augmented by generating the synthetic image. This study discusses a fast and real-like image synthesis approach, namely depthwise boundary equilibrium generative adversarial network (DepthwiseBEGAN). DepthwiseBEGAN was reduced memory load 70.11% in training processes compared to the conventional BEGAN. DepthwiseBEGAN synthetic images were inspected by measuring the Fréchet inception distance (FID) score with the real-to-real score equal to 4.3866 and real-to-fake score equal to 4.4674. Moreover, generated DepthwiseBEGAN synthetic images improve 22.59% accuracy of conventional CNN models.

Corresponding Author:

Zendi Iklima
Department of Electrical Engineering, Universitas Mercu Buana
Jakarta, Indonesia
Email: zendi.iklima@mercubuana.ac.id

1. INTRODUCTION

SARS-Cov-2 or known as coronavirus disease (COVID-19) was first identified in Wuhan, China [1] and designated by World Health Organization (WHO) as a global epidemic [2] because it has infected all corners of the world. At the end of February 2021, there were 116,521,281 confirmed cases, with 116,521,281 active cases and 2,589,548 cases of death. Meanwhile, the spread of COVID-19 in Indonesia also continues to increase, with February 2021 confirmed 1,379,662 cases, with 14,518 active cases, and 37,266 death cases [3]. Therefore, steps are needed to prevent, detect, and control the spread of the COVID-19 virus.

Early detection is one way to break the chain of the COVID-19 virus spread. When a patient is known to be positive for COVID-19, he will undergo a quarantine period so that the chain of spread can be broken by tracking the people who had interacted with the patient. One of the tests that can be done in early detection of COVID-19 virus is by conducting a test called reverse transcription-polymerase chain reaction (RT-PCR) to find out whether a patient is indicated as positive or negative from the COVID-19 virus infection. As time goes by, the pandemic condition continues even though the data on the spread is still high. Unfortunately, RT-PCR testing has less accurate results (40% to 60%) [4], [5] in determining positive or negative status of being infected with the COVID-19 virus [6], [7].

Alternative methods to detect the spread of the COVID-19 virus are through chest screening, namely computer tomography (CT) scan and chest X-ray (CXR) [8]. The resulting image of CT or X-ray has higher sensitivity than testing using RT-PCR. Thus, many automation systems have been developed in CT and X-rays image processing [6]. Displaying images via CT can detect COVID-19 virus infection. However, the procedure for testing via CT is costly, as well as age-restricted and forbidden for pregnant women because of the radiation. Therefore, studies in [9] use the CXR process that can be used for easy, fast, and inexpensive testing. Various deep learning methods are used as an automation system for CXR image processing to support the detection process for the COVID-19 virus infection. The convolutional neural network (CNN) method obtained accuracy in positive/negative CXR classification of 98.50% [10], while the deep neural network (DNN) method received accuracy in the positive/negative CXR classification of 98.08% [4]. CXR classification using deep CNN (DCNN) has an accuracy of 87.3% [11], while classification using generative adversarial networks (GANs) have an accuracy of 95% [12]. The depthwise separable convolution (DSC) network has an accuracy of 99.50% [13], and the COVIDX-Net has an accuracy of 91% [14]. On the other hand, public dataset images of CT or X-ray are limited. However, the classification method can be tuned by augmenting the dataset.

The deep learning approach is an interesting topic to develop an automation system to diagnose CXR images of the COVID-19 virus. LightCovidNet, which consists of a lightweight CNN (LW-CNN) and GANs with a frontal CXR dataset of 446 (resolution 1024X1024 pixels), with network filters to 841.771 parameters successfully trained the data with an accuracy of 96.97%. The separable convolution technique can reduce the memory load when processing training data (training data) 27 times more efficiently than conventional CNN, which consists of 23,567,299 parameters [15]. CovidGAN which consists of CNN and auxiliary classifier GANs (AC-GANs) methods using 403 CXR datasets (14,000,000 parameters) increases the accuracy of conventional GANs data augmentation 85% to 95% [12]. Covid-Net using the DCNN method using 13.975 CXR datasets (CovidX dataset of 11,750,000 parameters) yields an accuracy of 93.3% [16]. Coro-Net using the DNN method using 125 CXR datasets (33,915,436 parameters) yields an accuracy of 95% [17]. RANDGAN (randomized GAN) is ANO-GANs, using 573 CXR datasets resulting in an accuracy of 71% [5]. GANs and ResNet18 used the 5863 CXR datasets resulting in an accuracy of 99% [18].

To improve the model performance of the classification methods, we proposed a new architecture called DepthwiseBEGAN in which combining depthwise separable convolution (DSC) and BEGAN. This approach proposes augmented synthetic images of COVID-19 CXR dataset using DCGAN, DeptwiseGAN, BEGAN, and DepthwiseBEGAN. To exhibit DepthwiseBEGAN reduces the training load while the synthetic images are generated. In this research, we also measured the quality of generated images using Fréchet inception distance (FID). Additionally, the improvement of the classification method using generated synthetic images as fake CXR datasets is presented in this paper. Several classification models are used, such as: ResNet18, ResNet34, ResNet50, and GoogleNet., such as ResNet18, ResNet34, ResNet50, and GoogleNet.

2. RESEARCH METHOD

2.1. Depthwise separable convolution

CNN is a subclass of DNNs that can solve vision problems. CNN consists of the primary process, namely features extraction and fully connected layer. The convolutional layer is the fundamental layer of CNN that determines the characteristics of the image pattern as an input matrix to traverse through filters. By assuming $\left(x^{l}_{i^{l+1}}\right)$ as an input tensor which consist of triple index such as height (h^l), width (w^l), and depth (d^l). Spatial location of (h^l, w^l) utilized from bank filter of f. d^l is a receptive field in x^l. Therefore, the output of CNN layer can be denoted as [18], [19]:

$$y_{w^{l+1},h^{l+1},d} = \sum_{w=0}^{W} \sum_{h=0}^{H} \sum_{d=0}^{D} f_{w,h,d} \times x^{l}_{h^{l+1}+h,w^{l+1}+w,d} \tag{1}$$

where $D_f x D_f$ is an input matrix per M channels, therefore, total parameters in a kernel formulated as (2).

$$D_k^2 \times D_c^2 \times M \times N \tag{2}$$

CNN models with high-resolution images require more memory allocation due to the number of convolutional parameters in the kernel that must be calculated as vectors. Therefore, several CNN models can be simplified by reducing convolutional trainable parameters. DSC is a model that effectively reduces the number of convolutional parameters and matrix calculations with the limitations of precision. Conventional CNN utilized a convolutional kernel with the same input channel so that the matrix calculation is carried out by convolution channels per N channel, with the total parameters shown in (4) [20]. DSC consists of 2 convolution processes, namely depthwise convolution and pointwise convolution. Based on (1), DSC to distribute feature learning, namely depthwise and pointwise [2], formulated as [18], [19], [21]:

$$y_{w^{l+1},h^{l+1},d} = \sum_{d=0}^{D} f_d \times \sum_{h=0}^{H} \sum_{w=0}^{W} f_{w,h} \times x^{l}_{h^{l+1}+h,w^{l+1}+w} \tag{3}$$

where f_d is a pointwise in which 1×1 convolution layer. Figure 1 represent DSCN parameters utilized for computing the parameter load for each training process.

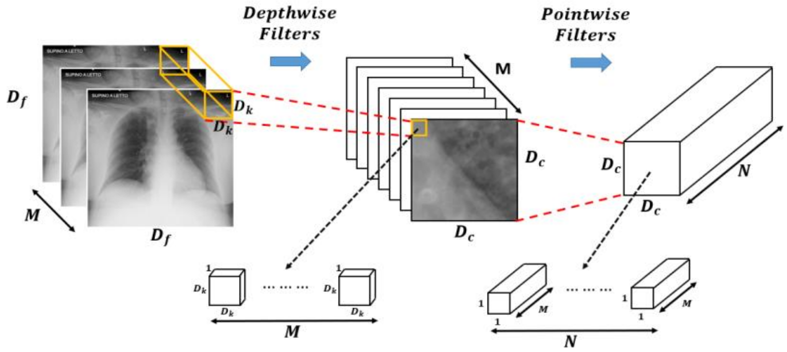


Figure 1. DSCN parameters in CXR of COVID-19

The number of parameters in the DSCN per 1 channel is denoted in (4). However, the total parameters in the DSCN are the total parameters in the depthwise convolution and pointwise convolution calculated as shown in (4). Compare to the (2), a comparison of the number of parameters on the CNN standard with the DSCN shown in (5) [22]. For example, if a convolution has $N = 1024$ and $D_k = 3$, there will be a reduction in the convolution parameters in the training process by 0.112 or 0.888 which mean DSC is able to reduce the training load than conventional CNN.

$$(D_c^2 \times M)(D_k^2 \times N) \tag{4}$$

$$\frac{pDSCN}{pCNN} = \frac{(D_c^2 \times M)(D_k^2 \times N)}{(D_k^2 \times D_c^2 \times M \times N)} = \frac{(D_k^2 \times N)}{(D_k^2 \times N)} \tag{5}$$

2.2. Deep convolution generative adversarial network

GANs were introduced in 2014 by Goodfellow which states that GANs consist of two networks namely generator network (G) and discriminator network (D). Both models are trained using the mini-max concept. Generator model $G(x;\theta_g)$, can train noise data label on $P_z(z)$ distribution data against x label or real data label. Discriminator model $D(x;\theta_d)$, trains the P_g distribution data be able to estimate the distribution data $P_{Data(x)}$ [23]. The data distribution $P_{Data(x)}$ is a positive image of CXR COVID-19 [24]. Generator model $G(z;\theta_g)$, minimizes the probability data distribution in the fake dataset $z{\sim}P_z$ that formulated as (6) [25]:

$$\min_{G} \mathrm{V}(G) = \mathbb{E}_{z\sim P_z}\left[log\left(1 - D\big(G(z)\big)\right)\right] \tag{6}$$

As shown in (6) shows the generator networks able randomized noise data distribution $P_z(z)$ to fool discriminator network which labelled in data distribution of $z{\sim}P_{Data(x)}$. Thus, Discriminator model $D(x;\theta_d)$ maximize the probability data distribution in $P_{Data(x)}$ formulated as (7) [25].

$$\max_{D} \mathrm{V}(G) = \mathbb{E}_{x\sim P_{data(x)}}\left[log\big(D(x)\big)\right] \tag{7}$$

Therefore, GANs mini-max term based on (6) and (7) can be formulated as (8) [25]:

$$\mathcal{L}_{adv} = \min_{G} \max_{D} V(G, D) = \mathrm{E}_{x\sim P_{data(x)}}[\log(D_{src}(x))] + \mathrm{E}_{z\sim P_z}[\log(1 - D_{src}(G(z)))] \tag{8}$$

where $\mathbb{E}(.)$ denotes as network expectation given by generator network and discriminator network, $V(G, D)$ is a training criterion of discriminator network given by generator network, where $D: x \rightarrow \{D_{src}(x), D_{cls}(x)\}$ denotes discriminator probability distributions over both source and its labels. Both discriminator network $D(.)$ and generator network $G(.)$ is able to be optimized using given objective functions formulated as [23]:

$$\mathcal{L}^{r}_{cls} = \mathbb{E}_{x\sim P_{data(x)}}[-logD_{cls}(x)] \tag{9}$$

$$\mathcal{L}^{f}_{cls} = \mathbb{E}_{z\sim P_z}[-logD_{cls}(G(z))] \tag{10}$$

$$\mathcal{L}_{rec} = \mathbb{E}_{x\sim P_z}\left[\left\|x - G_x(G(z))\right\|_1\right] \tag{11}$$

where λ_{cls}denotes as hyperparameters that optimize domain classification loss of discriminator $(\mathcal{L}^{R}_{cls})$ and generator $(\mathcal{L}^{f}_{cls})$. λ_{res} denotes as hyperparameters that optimize reconstruction loss $(\mathcal{L}_{res})$ that adopt L_1 normalization [23]. $\mathcal{L}_{res}$ translated $G(z)$ into $x\sim P_z$which mean that the generator $G_x(.)$ tries to reconstruct fake labels into real labels.

2.3. Depthwise boundary equilibrium GAN

Figure 2 shows DepthwiseBEGAN architecture given an input image shape (32, 32, 3). DSConv is depthwise separable convolution layer which contains depthwise layer and pointwise layer. Down-sample size transformed from given input shape 32×32 into 4×4 and 8×8.

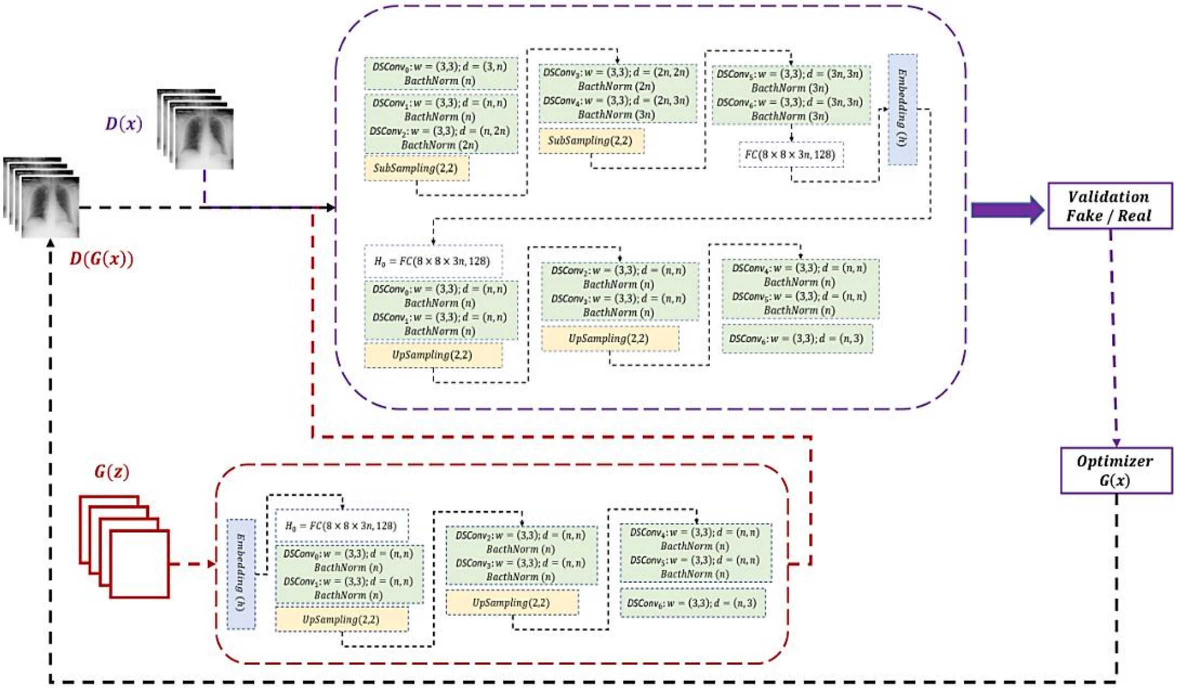

$D(x)$
$D(G(x))$
$DSConv_0: w = (3,3); d = (3, n)$
$BacthNorm\ (n)$
$DSConv_1: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_2: w = (3,3); d = (n, 2n)$
$BacthNorm\ (2n)$
$SubSampling(2,2)$
$DSConv_3: w = (3,3); d = (2n, 2n)$
$BacthNorm\ (2n)$
$DSConv_4: w = (3,3); d = (2n, 3n)$
$BacthNorm\ (3n)$
$SubSampling(2,2)$
$DSConv_5: w = (3,3); d = (3n, 3n)$
$BacthNorm\ (3n)$
$DSConv_6: w = (3,3); d = (3n, 3n)$
$BacthNorm\ (3n)$
$FC(8 \times 8 \times 3n, 128)$
$Embedding\ (h)$
$H_0 = FC(8 \times 8 \times 3n, 128)$
$DSConv_0: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_1: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$UpSampling(2,2)$
$DSConv_2: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_3: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$UpSampling(2,2)$
$DSConv_4: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_5: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_6: w = (3,3); d = (n, 3)$
Validation Fake / Real
Optimizer $G(x)$
$G(z)$
$Embedding\ (h)$
$H_0 = FC(8 \times 8 \times 3n, 128)$
$DSConv_0: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_1: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$UpSampling(2,2)$
$DSConv_2: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_3: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$UpSampling(2,2)$
$DSConv_4: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_5: w = (3,3); d = (n, n)$
$BacthNorm\ (n)$
$DSConv_6: w = (3,3); d = (n, 3)$

Figure 2. Architecture of DepthwiseBEGAN of COVID-19 CXR images

The term of equilibrium to balance auto-encode real dataset $(x; \theta_d)$ and discriminate $G(z; \theta_g)$ which equalized as (12) [26]:

$$\mathbb{E}[\mathcal{L}(G(z))] = \gamma\mathbb{E}(\mathcal{L}(x)) \tag{12}$$

where γ denotes as diversity ratio, which maintains the equilibrium using proportional control theory ($k_t \in [0,1]$). Based on (12), boundary equilibrium GAN (BEGAN) represent as an objective function as (13)-(15) [27], [28]:

$$\mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) \tag{13}$$

$$\mathcal{L}_G = \mathcal{L}(G(z_G)) \tag{14}$$

$$k_{t+1} = k_t + \lambda_k (\mathrm{g}\mathcal{L}(x) - \mathcal{L}(G(z_G))) \tag{15}$$

where λ_k is a proportional gain for k, $\mathcal{L}(x) = \mathbb{E}_{x\sim P_{data(x)}}[||D_{cls}(x) - x\|_1]$ denotes as auto-encoder loss $L_1$of data distribution $x\sim P_{Data(x)}$, and $\mathcal{L}(G(z)) = \mathbb{E}_{x\sim P_z}\left[\left\|D_{cls}(G(z)) - G(z)\right\|_1\right]$ denotes as auto-encoder loss $L_1$of data distribution $z\sim P_z$. Essentially, (16) shows a form of closed-loop feedback control, in which k_t is adjusted at each step $t+1$. The equilibrium constraint manages the training process to yielding to $\mathcal{L}(x) > \mathcal{L}(G(z))$. Therefore, convergence global measurement of the equilibrium which denotes as (16) [27], [28].

$$\mathcal{M}_{global} = \mathcal{L}(x) + |\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G)) \tag{16}$$

2.4. Fréchet inception distance

Fréchet inception distance (FID) is utilized as a metric to assess image quality of GANs which approximates the distribution of fake generated images $D_{cls}(G(z_G))$ with the distribution of real images $D_{cls}(x)$ that were used to train the generator as multivariate Gaussians as (17) [29]:

$$\left\|\mu_r - \mu_g\right\|^2 + Tr\left(\textstyle\sum_r + \sum_g - 2\sqrt{\sum_r\sum_g}\right) \tag{17}$$

where $X_r\sim(\mu_r, \sum_r)$ denotes as mean of 2048-dimensional activation and $X_g\sim(\mu_g, \sum_g)$ denotes the covariance of 2048-dimensional activation which extracted from pre-trained Inception-v3 model. Our model transforms data distribution of real $\mathbb{E}_{x\sim P_{data(x)}}$ and fake $\mathbb{E}_{x\sim P_z}$into 32×32, 64×64, 128×128, and 256×256 image dimension.

3. RESULTS AND DISCUSSION

The training process was performed using cloud instance Intel(R) Xeon(R) CPU @ 2.30 GHz, high-memory VMs, 2xvCPU, 25 GB RAM, and GPU using NVIDIA P100/T4, peripheral component interconnect express (PCI Express) 16 GB. The training process consists of three schemas. The first schema was trained a conventional DCGAN, BEGAN, and DepthwiseGANs using particular dataset distributions. This schema generates real-like fake images or generate synthetic CXR image. The second schema was calculated the quality of augmented images divides into several batches of random images. The third schema was utilized to test whether the augmented images can be classified using particular classification method such as CNN.

3.1. Data distributions and hyperparameters

This paper represents three kind of datasets such as MNIST dataset, CelebA dataset, and COVID-19 CXR dataset. GAN model trained on 60K MNIST images, 24K CelebA images, and 5.4K CXR images to generate realistic image synthesis. This proposed approach trained both generator and discriminator network using Adam with an initial learning rate $\alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.999$, proportional gain ($\lambda_k = 0.7$) [30], varied image transformation 32 to 256. The hyperparameters in Table 1 show that DepthwiseGANs can propose data augmentation to generate synthetic CXR image using randomized noise inputs.

Based on Table 1, the hyperparameters engages performances among GAN types, especially in image-to-image translation containing DCGAN, DepthwiseGAN, BEGAN, and DepthwiseBEGAN. Figure 1 shows the generator and discriminator architecture as a convolutional feature extraction which down-sampled in 4x4 and 8x8. DCGAN has 7.12 million trainable parameters, DepthwiseDCGAN has 0.76 million trainable parameters, BEGAN has 8.44 million trainable parameters, and DepthwiseBEGAN has 2.23 million trainable parameters. Additionally, this research compares the hyperparameters combination shown in Table 1 to analyze the model performance based on generator loss ($\mathcal{L}_G$), discriminator loss ($\mathcal{L}_D$), and the execution time.

3.2. DepthwiseBEGAN training performance

Based on (11) until (15), generator loss ($\mathcal{L}_G$) and discriminator loss ($\mathcal{L}_D$) calculated in epoch=25, filters=64, image resolution 32x32, random noise=48, and down-sampled size 4x4. Then, Table 2 represent generator loss ($\mathcal{L}_G$) and discriminator loss ($\mathcal{L}_D$) of DCGAN and DepthwiseGAN, which consists of following datasets such as MNIST, CelebA, and CXR dataset. Table 2 represent DCGAN and DepthwiseGAN training metrics.

Table 1. Hyperparameters

Description	Parameters
Model	DCGAN [30], DepthwiseGAN [23], BEGAN [29], DepthwiseBEGAN
Dataset	MNIST, CelebA [29], CXR
Down-sampled Size	8x8 [29], 4x4
Filters/Batch Size	4, 32, 64
Noises Input	48
Epoch	25
Image Resolution	32x32, 64x64, 128x128 [29], 256x256

Table 2. Loss of DCGAN and DepthwiseGAN using particular datasets

Model	Dataset	$\mathcal{L}_G$	$\mathcal{L}_D$	Exec. Time (Minutes)
DCGAN	MNIST	5.2513	0.1130	33.1355
	CelebA	2.3994	0.7012	37.1142
	CXR	1.7703	0.8355	29.7863
DepthwiseGAN	MNIST	3.7615	0.2940	24.6837
	CelebA	2.5109	0.6646	21.3387
	CXR	2.3558	0.6418	17.5

Table 2 shows generator loss ($\mathcal{L}_G$), discriminator loss ($\mathcal{L}_D$), and execution time of both models DCGAN and DepthwiseGAN. Generator loss ($\mathcal{L}_G$) and discriminator loss ($\mathcal{L}_D$) of both models DCGAN and DepthwiseGAN closely fit, but the execution time of DepthwiseGAN is lower than DCGAN which follows the number reduction of trainable parameters. Table 3 shows generator loss ($\mathcal{L}_G$), discriminator loss ($\mathcal{L}_D$), and execution time of both models DCGAN, DepthwiseGAN, BEGAN, and DepthwiseBEGAN in epoch=25, filters=(4 and 32), image resolution=(64x64, 128x128, and 256x256), and random noise=48.

Table 3. Loss of DCGAN, DepthwiseGAN, BEGAN, and DepthwiseBEGAN using CXR datasets

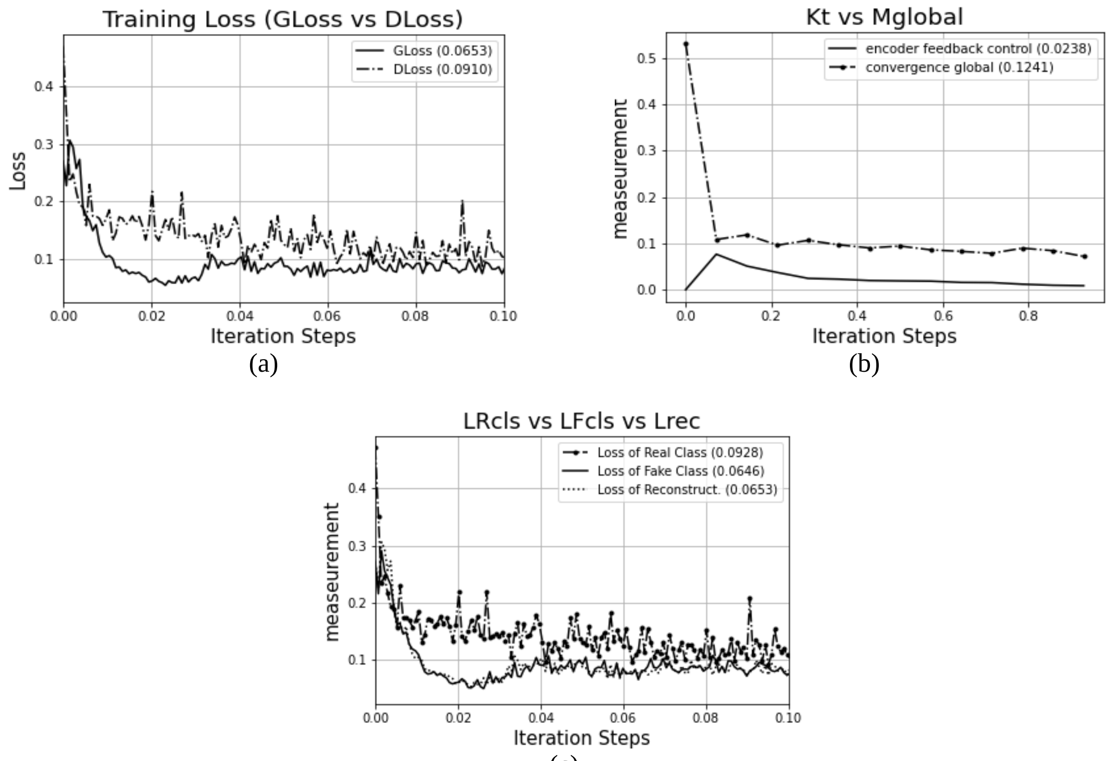
Model	Filters	Image Resolution	$\mathcal{L}_G$	$\mathcal{L}_D$	Exec. Time (Minutes)
DCGAN	32	64x64	1.4017	0.6744	49.7863
DepthwiseGAN	32	64x64	1.5538	0.5188	20.2133
BEGAN	32	64x64	0.0451	0.0885	76.5556
DepthwiseBEGAN	32	64x64	0.0465	0.0811	22.8859
BEGAN	32	128x128	0.0635	0.0789	132.3350
DepthwiseBEGAN	32	128x128	0.0643	0.0799	48.7891
BEGAN	4	256x256	0.0797	0.0989	186.4425
DepthwiseBEGAN	4	256x256	0.0785	0.0965	117.4362

Based on Table 3 DepthwiseBEGAN execution time was faster than BEGAN execution time in particular filers and image resolution. Meanwhile, generator loss ($\mathcal{L}_G$) and discriminator loss ($\mathcal{L}_D$) closely fit in the training stage. DepthwiseBEGAN is able to augment synthetic images with 256x256 pixels. However, the number of filters was reduced because of GPU limitations.

DepthwiseBEGAN is shown in Figure 3. Generator loss ($\mathcal{L}_G$) and discriminator loss ($\mathcal{L}_D$) of DepthwiseBEGAN shown in Figure 3(a), proportional control (k_{t+1}), and convergence global ($\mathcal{M}_{global}$) of DepthwiseBEGAN shown in Figure 3(b), and domain classification loss of discriminator ($\mathcal{L}_{cls}^{r}$), domain classification loss of generator ($\mathcal{L}_{cls}^{f}$), and reconstruction loss ($\mathcal{L}_{res}$) shown in Figure 3(c).

3.3. DepthwiseBEGAN performance measurement

Based on Figure 2 DepthwiseBEGAN performed in filter size=4, image resolution=256x256, epoch=25, and random noise=48. Image quality of GANs is able to be assessed by measuring FID, which approximates the distribution of real-to-real images (RR), the distribution of fake-to-real (FR) images, and the distribution of fake-to-fake (FF) images. Measurement of FID was captured in Table 4 for each 12K iteration steps in DepthwiseBEGAN training process.



(a) (b)

(c)

Figure 3. DepthwiseBEGAN (a) training loss $\mathcal{L}_G$ and $\mathcal{L}_D$, (b) $\mathcal{M}_{global}$ and k_{t+1}, and (c) domain loss $\mathcal{L}^r_{cls}, \mathcal{L}^f_{cls}$, and $\mathcal{L}_D$

Table 4. FID score of BEGAN and DepthwiseBEGAN

Model	Batch	FID RR	FID FF	FID FR
BEGAN	1	4.3866	4.6633	4.4098
DepthwiseBEGAN				4.4674
BEGAN	5	17.2621	12.9109	20.9808
DepthwiseBEGAN				25.9938
BEGAN	20	59.2829	39.3068	69.5281
DepthwiseBEGAN				77.2037
BEGAN	50	104.9618	68.2335	146.6602
DepthwiseBEGAN				157.7203

Table 4 measures FID with several batches containing 1, 5, 20, and 50 images. GANs evaluate by propagating the distribution of RR or FR or FF using pre-trained Inception-v3. A proportional FID value while measuring the same image equal to zero. By calculating random image in batch size is 1, FID value of RR equal to 4.3866, FID value of FR equal to 4.4098 and FID value of FF equal to 4.6633. Synthetic images augmented by DepthwiseBEGAN is shown in Figure 4. Therefore, Figure 4(a) shows synthetic images of CXR with normal label which augmented by DepthwiseBEGAN. Figure 4(b) shows synthetic images of CXR with normal bacteria/virus label which augmented by DepthwiseBEGAN.

Synthetic images of fake generated images $D_{cls}\big(G(z)\big)$ in CXR dataset has been augmented, which distributes normal label of train images equal to 12.49K images, the normal label of validation images equal to 3.75K images, the normal label of test images equal to 7.13K images, the virus label of train images equal to 12.98K images, the virus label of validation images equal to 3.49K images, the virus label of test images equal to 6.49K images, the bacteria label of train images equal to 20.01K images, the bacteria label of validation images equal to 4.99K images, and the bacteria label of test images equal to 7.49K images.

Augmented CXR images trained using several CNN models such as RestNet18 [27] , ResNet-50 [16], and VGG19 [28]. The real and fake generated data distribution with input resolution 128x128 trained using Adam optimizer with an initial learning rate $\alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.999$ [30] and 50 iterations. In order to represent the performance of particular CNN models, this paper assigned the accuracy, specificity, sensitivity, positive predictive value (PPV), and negative predictive value (NPV), can be formalized as [27]:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{18}$$

$$Sensitivity = (TP)/(TP + FN) \tag{19}$$

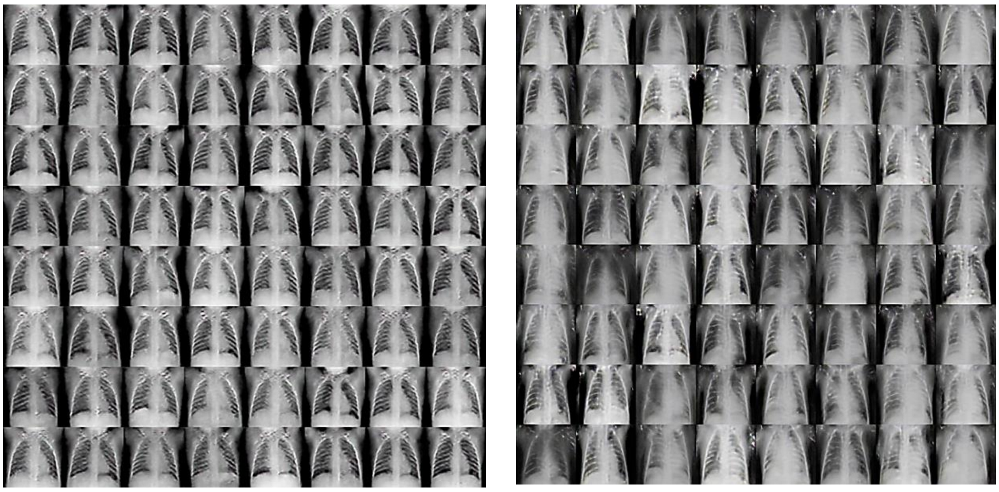
$$Specificity = (TN)/(TN + FP) \tag{20}$$

$$PPV = (TP)/(TP + FP) \tag{21}$$

$$NPV = (TN)/(TN + FN). \tag{22}$$

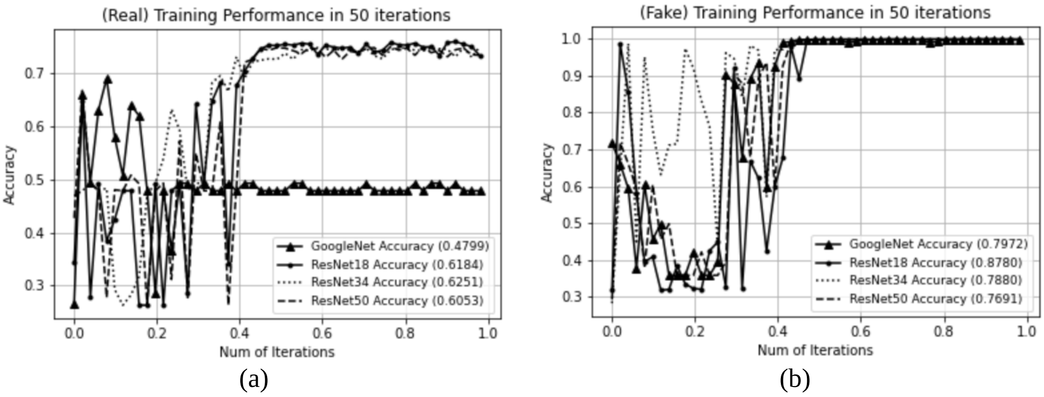
Based on (19) until (22), sensitivity and specificity are defined for a domain of binary classification. Sensitivity determines whether the ‘virus’ label meets the condition in TP divided by TP and FN. Specificity determines the virus label does not meet condition means FP divided by FP and TN. Positive predictive value (PPV) is determines the ‘virus’ label meets condition of positive direction means TP divided by TP and FP. Negative predictive value (NPV) is determines the ‘virus’ label meets condition of negative direction means TN divided by TN and FN. Figure 5 shows CNN models utilized to classify the CXR images based on the following labels, namely normal label, bacteria label, and virus label. Figure 5(a) represents CNN training accuracy using real CXR datasets and Figure 5(b) represents CNN training accuracy using generated or fake CXR datasets.

Figure 5 represent the CNN training accuracy using real and fake CXR dataset. Based on (18) until (22) the confusion matrix has calculated. The confusion matrix of the following CNN models was calculated using 100 images from particular sources which shown in Table 5.



(a) (b)

Figure 4. Synthetic images augmented by DepthwiseBEGAN (a) normal label and (b) bacteria/virus label



(a) (b)

Figure 5. CNN training accuracy using (a) real and (b) fake CXR images

Table 5. CNN Confusion matrix

Data Dist.	Models	Labels	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
Real	GoogleNet	Normal	64.29	75.00	50.00	84.38
$D_{cls}(x)$		Virus	43.75	64.71	36.84	70.97
		Bacteria	40.00	83.33	64.54	67.57
Fake Gen.	GoogleNet	Normal	100.00	97.22	93.33	100.00
$D_{cls}(G(z))$		Virus	93.75	94.12	88.24	96.97
		Bacteria	90.00	100.00	100.00	93.75
Real	ResNet-18	Normal	78.57	88.88	73.33	81.43
$D_{cls}(x)$		Virus	68.75	76.47	57.89	83.87
		Bacteria	70.00	93.33	87.5	82.35
Fake Gen.	ResNet-18	Normal	100.00	97.22	93.33	100.00
$D_{cls}(G(z))$		Virus	100.00	97.06	94.12	100.00
		Bacteria	90.00	100.00	100.00	93.75
Real	ResNet-34	Normal	80.00	88.57	75.0	91.18
$D_{cls}(x)$		Virus	72.22	78.13	65.00	83.33
		Bacteria	70.59	93.93	85.72	86.11
Fake Gen.	ResNet-34	Normal	100.00	97.22	93.33	100.00
$D_{cls}(G(z))$		Virus	100.00	96.97	94.44	100.00
		Bacteria	89.47	100.00	100.00	93.93
Real	ResNet-50	Normal	61.54	89.19	66.67	86.84
$D_{cls}(x)$		Virus	62.50	76.48	55.55	81.25
		Bacteria	76.19	86.21	80.00	83.33
Fake Gen.	ResNet-50	Normal	100.00	91.89	81.25	100.00
$D_{cls}(G(z))$		Virus	93.75	97.06	93.75	97.06
		Bacteria	85.71	100.00	100.00	90.63

4. CONCLUSION

One of the most common procedures to detect COVID-19 by chest screening using X-ray technology, CXR imaging accurately identifies whether a patient is infected with the COVID-19 virus or not. A computational approach proposed, such as CNN can classify CXR images within three labels: normal label, bacteria label, and COVID-19 label. Covid-Net trained the highest CXR images (14K images) to classify CXR images with 93% accuracy. Several methods utilized small CXR images to be trained on less than 10K images. Image synthesis method proposed to augment CXR images of COVID-19 within the goals to increase classification method accuracy.

In resolution 64×64, DCGAN trained to augmented CXR image synthesis within generator loss equal to 1.4017, discriminator loss equal to 0.6744, and execution time equal to 49.7863 minutes. DepthwiseGAN trained to augmented CXR image synthesis within generator loss equal to 1.5538, discriminator loss equal to 0.5188, and execution time equal to 20.2133 minutes. DepthwiseGANs have shortened execution time for a better-generated image by the discriminator. Quality of DepthwiseGANs improved by using the encoder-decoder model of GANs, namely BEGAN. BEGAN trained to augmented CXR image synthesis within generator loss equal to 0.0451, discriminator loss equal to 0.0885, and execution time equal to 76.5556 minutes. DepthwiseBEGAN trained to augmented CXR image synthesis within generator loss equal to 0.0465, discriminator loss equal to 0.0811, and execution time equal to 22.8859 minutes.

FID of DepthwiseBEGAN measured by comparing the number of image batches and the source of the image. Measurement of FID using one random image calculated fake-to-fake (FF) equal to 4.6633, real-to-real (RR) equal to 4.3866, and fake-to-real (FR) equal to 4.4674. Furthermore, generated DepthwiseBEGAN synthetic images improve 22.59% accuracy of conventional CNN models.

ACKNOWLEDGEMENTS

The author expresses gratitude to the Electrical Engineering Department and Research Center of Universitas Mercu Buana, Jakarta, Indonesia. Under the support, the authors have been completed this research.

REFERENCES

[1] D. Singh, V. Kumar, Vaishali, and M. Kaur, “Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks,” *European Journal of Clinical Microbiology & Infectious Diseases*, vol. 39, no. 7, pp. 1379–1389, Jul. 2020, doi: 10.1007/s10096-020-03901-z.

[2] T. Mahmud, M. A. Rahman, and S. A. Fattah, “CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization,” *Computers in Biology and Medicine*, vol. 122, Jul. 2020, doi: 10.1016/j.compbiomed.2020.103869.

[3] WHO, “WHO coronavirus (COVID-19) dashboard," WHO Health Emergency Dashboard. https://covid19.who.int/ (accessed Apr.

01, 2021).

[4] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, Jun. 2020, doi: 10.1016/j.compbiomed.2020.103792.

[5] S. Motamed, P. Rogalla, and F. Khalvati, "RANDGAN: Randomized generative adversarial network for detection of COVID-19 in chest X-ray," *Scientific Reports*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-87994-2.

[6] S. Hassantabar, M. Ahmadi, and A. Sharifi, "Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches," *Chaos, Solitons & Fractals*, vol. 140, Nov. 2020, doi: 10.1016/j.chaos.2020.110170.

[7] A. A. Ardakani, A. R. Kanafi, U. R. Acharya, N. Khadem, and A. Mohammadi, "Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks," *Computers in Biology and Medicine*, vol. 121, Jun. 2020, doi: 10.1016/j.compbiomed.2020.103795.

[8] J. Zhao, X. He, X. Yang, Y. Zhang, S. Zhang, and P. Xie, "COVID-CT-Dataset: A CT image dataset about COVID-19," *arXiv*, pp. 1–14, 2020.

[9] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207–1220, Aug. 2021, doi: 10.1007/s10044-021-00984-y.

[10] B. Sekeroglu and I. Ozsahin, "Detection of COVID-19 from chest X-ray images using convolutional neural networks," *SLAS Technology*, vol. 25, no. 6, pp. 553–565, 2020, doi: 10.1177/2472630320958376.

[11] Y. Zhong, "Using deep convolutional neural networks to diagnose COVID-19 from chest X-ray images," *arXiv: 2007.09695*, Jul. 2020.

[12] 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, doi: 10.1109/ACCESS.2020.2994762.

[13] M. Rahimzadeh and A. Attar, "A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2," *Informatics in Medicine Unlocked*, vol. 19, 2020, doi: 10.1016/j.imu.2020.100360.

[14] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-Net: A framework of deep learning classifiers to diagnose COVID-19 in X-Ray images," *arXiv preprint arXiv:2003.11055.*, Mar. 2020.

[15] M. A. Zulkifley, S. R. Abdani, and N. H. Zulkifley, "COVID-19 screening using a lightweight convolutional neural network with generative adversarial network data augmentation," *Symmetry*, vol. 12, no. 9, Sep. 2020, doi: 10.3390/sym12091530.

[16] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Scientific Reports*, vol. 10, no. 1, Dec. 2020, doi: 10.1038/s41598-020-76550-z.

[17] A. I. Khan, J. L. Shah, and M. M. Bhat, "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images," *Computer Methods and Programs in Biomedicine*, vol. 196, Nov. 2020, doi: 10.1016/j.cmpb.2020.105581.

[18] N. E. M. Khalifa, M. H. N. Taha, A. E. Hassanien, and S. Elghamrawy, "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: 2004.01184*, Apr. 2020.

[19] X. Xu *et al.*, "A deep learning system to screen novel coronavirus disease 2019 pneumonia," *Engineering*, vol. 6, no. 10, pp. 1122–1129, Oct. 2020, doi: 10.1016/j.eng.2020.04.010.

[20] E. Ihsanto, K. Ramli, D. Sudiana, and T. S. Gunawan, "Fast and accurate algorithm for ECG authentication using residual depthwise separable convolutional neural networks," *Applied Sciences*, vol. 10, no. 9, May 2020, doi: 10.3390/app10093304.

[21] N. K. Chowdhury, M. M. Rahman, and M. A. Kabir, "PDCOVIDNet: a parallel-dilated convolutional neural network architecture for detecting COVID-19 from chest X-ray images," *Health Information Science and Systems*, vol. 8, no. 1, Dec. 2020, doi: 10.1007/s13755-020-00119-3.

[22] K. KC, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, Oct. 2019, doi: 10.1016/j.compag.2019.104948.

[23] M. Ngxande, J.-R. Tapamo, and M. Burke, "DepthwiseGANs: Fast training generative adversarial networks for realistic image synthesis," in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, Jan. 2019, pp. 111–116, doi: 10.1109/RoboMech.2019.8704766.

[24] F. Munawar, S. Azmat, T. Iqbal, C. Gronlund, and H. Ali, "Segmentation of lungs in chest X-ray image using generative adversarial networks," *IEEE Access*, vol. 8, pp. 153535–153545, 2020, doi: 10.1109/ACCESS.2020.3017915.

[25] Y. Jiang, H. Chen, M. Loew, and H. Ko, "COVID-19 CT image synthesis with a conditional generative adversarial network," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 2, pp. 441–452, Feb. 2021, doi: 10.1109/JBHI.2020.3042523.

[26] Y. Li, N. Xiao, and W. Ouyang, "Improved boundary equilibrium generative adversarial networks," *IEEE Access*, vol. 6, pp. 11342–11348, 2018, doi: 10.1109/ACCESS.2018.2804278.

[27] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Applied Intelligence*, vol. 51, no. 2, pp. 854–864, Feb. 2021, doi: 10.1007/s10489-020-01829-7.

[28] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, Jun. 2020, doi: 10.1007/s13246-020-00865-4.

[29] B. Huang, W. Chen, X. Wu, C.-L. Lin, and P. N. Suganthan, "High-quality face image generated with conditional boundary equilibrium generative adversarial networks," *Pattern Recognition Letters*, vol. 111, pp. 72–79, Aug. 2018, doi: 10.1016/j.patrec.2018.04.028.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, 2016.

BIOGRAPHIES OF AUTHORS

Zendi Iklima received the M.Sc. degree in Software Engineering from Beijing Institute of Technology in 2018. His research interests include robotics, deep learning, and cloud computing. He has an experience as the Chief Technology Officer of Diaspora Connect Indonesia. Indonesian Diaspora Connect is one of the controversial start-up companies in 2018. The idea of connecting Indonesia is to help the Indonesian government find out about citizens who are studying or living abroad which integrates with artificial intelligence. Based on the idea, he achieved 1st place in ALE Hackathon Competition in 2018 which held by Alcatel-Lucent (Jakarta). He can be contacted at email: zendi.iklima@mercubuana.ac.id.

Trie Maya Kadarina received her bachelor's degree in Electrical Engineering from Institut Teknologi Nasional in 2001. She received her master's degree in Biomedical Engineering from the Department of Electrical Engineering, Institut Teknologi Bandung, in 2005. Currently, she is a lecturer in the Department of Electrical Engineering at Universitas Mercu Buana. Her research interests include biomedical instrumentation, electronics and control systems, machine learning, and internet of things. She can be contacted at email: trie.maya@mercubuana.ac.id.

Eko Ihsanto received the undergraduate and Doctorate degree from Electrical Engineering Department University of Indonesia. His research interests include embedded design, signal processing and deep learning. He can be contacted at email: eko.ihsanto@mercubuana.ac.id.