**УДК 61:614.4:773.713.4** https://doi.org/10.31713/vt4202013

**Yaroshchak S. V., Candidate of Engineering (Ph.D.), Associate Professor, (0000-0001-9576-2929, s.v.yaroshchak@nuwm.edu.ua), Smaida Mahmoud (0000-0002-5552-2768, m.e.smaida@nuwm.edu.ua)** (National University of Water and Environmental Engineering, Rivne)

# GENERATING REALISTIC MEDICAL IMAGES USING DCGAN FOR ENHANCING COVID-19 CLASSIFICATION

**In image classification, the most important factors are the amount of data, especially in medical images. However, obtaining medical images becomes big challenge. In this paper, we present Deep Convolutional Generative Adversarial Network (DCGAN) method that generate synthetic medical images. In addition, we will use our model that we used to classify eye diseases to identify Coronavirus. The model will be tested with and without synthetic medical images in order to measure the model accuracy. Three types of chest diseases (X-Ray images), Covid-19, Pneumonia and Normal will be used as a dataset. Our method is demonstrated on a limited dataset of chest disease (155 Covid-19, 104 Pneumonia and 168 Normal). Firstly, we exploit DCGAN to generate synthetic medical images, then we utilize GMD method for chest diseases classification in order to classify Coronavirus, finally, training our method using all images, original data and synthetic medical images, then compare performance.**

**The accuracy of the model had improved significantly from 86.32% in training set and 85.50% in validation set, to 95.45% in training set and 86.42% in validation set. We suggest that this work can be applied to other image classification models such as Vgg16, Inception v3, ResNet to enhance the accuracy. Number of epochs is very important to generate high quality images.**

***Keywords*: Covid-19; Deep learning; DCGAN; medical images.**

## 1. Introduction

The beginning of Covid-19 disease was from Wuhan, China, in December 2019, as this disease spread every corner of the world within a short period of time [1]. On March 11, 2020, the World Health Organization declared this disease a pandemic. In July 2020, the number of cases in the world reached about 12 million cases worldwide, and the number of deaths in that period reached approximately 562039. Thus, it

became important to detect infected persons early and isolate them [2; 3].

One of the methods used to detect this virus is a PCR device, which is a procedure to collect samples from the nose or throat. However, this device has a high false negative rate, moreover it is not available in many countries of the world. Thus, medical images such as a CT scan or an X-ray image may be the best alternative to detect this virus.

One of the main challenges in the field of medical imaging is how to deal with small data sets, especially when we use supervised learning to classify images. In medical imaging assignments, explanatory reports are made by specialists with experience in his or her specialty, and most annotations from medical images are time consuming. Although, some of medical data sets are available online, and major challenges have been announced, most data sets remain limited in size and only apply to specific medical problems. The collection of medical data is a complex and expensive procedure that requires the cooperation of researchers and radiologists [4].

Many researchers are trying to overcome this challenge by using augmentation diagrams, including making some adjustments to the images of the data set such as rotation, cropping and size. Using such redundant data to improve network training has become a standard procedure for computer vision tasks, recently one of the most important method used is GANs.

Generative Adversarial Networks (GANs), are a type of machine learning network that Ian Goodfellow and colleagues invented in 2014. Two neural networks compete with each other in a game (meaning game theory, often but not always in the form of a zero-sum game). The aim of it is to train in creating fabricated data similar to real data, which are difficult for a human or mechanical observer to distinguish between them. This technique learns to generate new data with the same statistical properties of a training set. For example, a GAN trained in photographs can create new images that look real to human observers, and have many photorealistic characteristics. Although it was originally proposed as a form of the generative model of unsupervised learning, GANs have also proven useful for semi-supervised learning, fully supervised learning, and reinforcement learning. At the 2016 symposium, AI expert Yann LeCun described GANs as "the coolest idea in the field of machine learning in the past 20 years" [5].

The Generative Adversarial Network consists of two separate neural networks, Generator (G) and Discriminator (D) constantly competing

against each other, the generator, which tries to generate examples that seem the real data, it works to deceive the discriminator, while the role of discriminative network is to distinguish between the real and fake data. a vector of random number will be feeds to the generator, and the outputs are unreal synthetic data denoted by G (z), the discriminator feeds real data and the output of generator. The output of the discriminator is the possibility whether the entered data is real or fake, the output denoted by D (x). The output D (x) represents the probability that x will be a real image if it is 1, which means 100% is a real image, and the output is 0 which means that the image cannot be real.

During training, the goal of creating the Generator is to try to create real images to deceive the Discriminator. The goal of Discriminator is to separate the Generated images from the real images as closely as possible.

What is the outcome of the final match? In the most perfect case, Generator can create a real and fake G (z) images. For D it is difficult to determine whether the image generated by G is real so D (G (z)) = 0.5.

In this paper, we investigate the problem of lack of medical images such as chest diseases for image classification. We propose Generative Adversarial networks (GANs) to increase our dataset in order to improve the accuracy of classification using our model (Glaucoma, Myopia and Diabetic retinopathy model). The model based on CNN for image classification.

In this way, our goal is achieved and we obtain the generative model G, which can be used to create images in order to increase our dataset [7].

**1.1. How GAN is work:**

The above is just a rough overview of the basic principles of GAN, how can it be described in mathematical language? Here is a straightforward excerpt from the formula from Ian Goodfellow's paper, which called objective function of GAN [6]:

$$\min_G \max_D V(D,G) = E_{x\sim pdata(x)}[\log(D(x)] + E_{z\sim pz(z)}[\log(1-D(G(z)))]. \quad (1)$$

You only need to analyze this formula:

- The complete formula consists of two periods. X represents the real image; Z represents the noise input to the G network.
- G (z) is a fake image generated by the generator G.
- D (x) is the probability of that X came from the real data (0~1), The discriminator should classify a real image as real. And this value should be close to 1.

- Purpose of G as mentioned above. D (G (z)) is the probability that a D network governs whether the image generated by G is real. In other words, G wants D (G (z)) to be as large as possible (close to 1), and V (D, G) will be smaller at this time. So, we see that the forward tag of the formula is min_G.
- Purpose of D: is simply a classifier. It tries to distinguish real data from the data created by the generator. train D to classify real images as real, D(x) should be close to 1. Train D to classify fake images as fake, D (G (z)) should be close to 0.

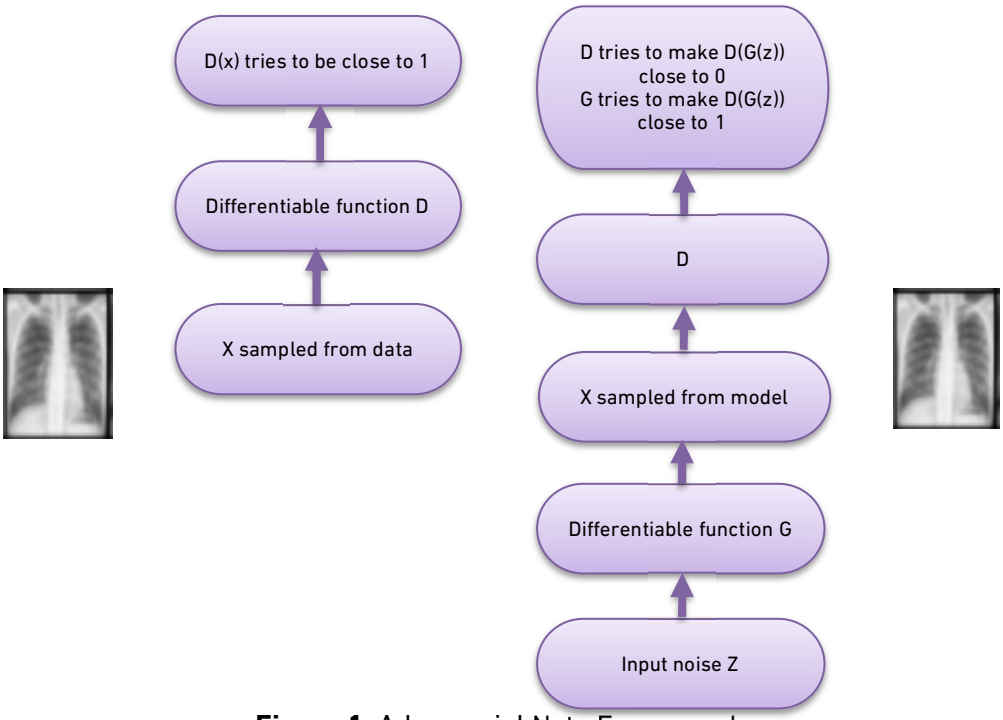



**Figure 1.** Adversarial Nets Framework

**1.2. Training process in GAN:**

Since the generative model and the discriminant model in the generative adversarial network are completely independent models and cannot be trained at the same time, separate repetitive training is used. For the discriminant, the label for the true sample is marked with the number 1, and the label for the generated sample is marked as 0, regardless of the quality of the sample generated.

Updating the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)]. \quad (2)$$

Updating the generator by descending its stochastic gradient:

$$\nabla_{\theta g} \frac{1}{m} \sum_{i=1}^{m} log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right). \quad (3)$$

We train D with the hope that V (G, D) is close to 1, so we add a gradient (ascending). When training G in the second step, V (G, D) is as small as possible close to 0, so the gradient is subtracted. And then the entire training process rotates.

**1.3. Some types of GANs**

There have been many types of GAN since its first appearance in 2014, including Convolutional GAN (CGAN), Deep Convolutional GANs (DCGANs), Least Squares GANs(LSGAN), Boundary Equilibrium GANs(BEGAN), StackGAN, InfoGANs, Wasserstein GANs (WGAN) and DiscoGANs. The DCGAN that has been used in this paper will be explained briefly [7; 8; 9; 10; 11].

**1.3.1. Concept of DCGAN**

DCGAN is Deep Convolutional Generative Adversarial Network, the same as GAN, so we won't repeat it here. It only replaces the G and D above with two convolutional neuronal networks (CNN). But the change is not directly enough, DCGAN made some changes to the structure of the convolutional neural network to improve sample quality and convergence velocity. These changes include:

- Instead of using pooling layers use stride convolutions (discriminator) and fractional-stride convolutions (generator).
- Use batch normalization in both D and G.
- Remove the FC layer and make the network a fully convolutional network.
- The G ReLU network is used as the activation function, and the last layer uses the tanh.
- Use LeakyReLU as the activation function in the D network.

Figure 2 shows G-Network in DCGAN [6].

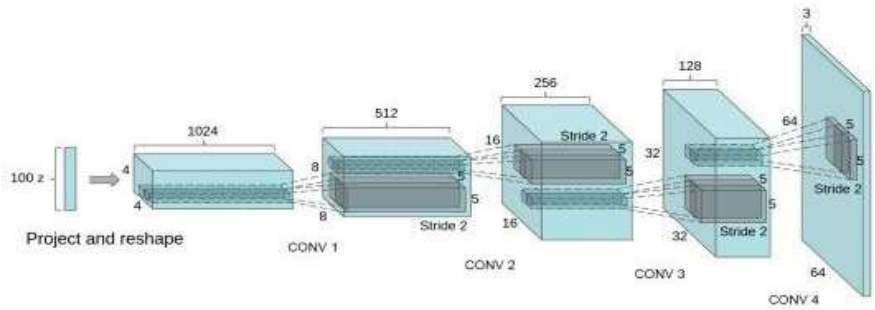



**Figure 2.** G-Network in DCGAN

## 2. Related work

Several researches have been published in the study of GAN. Most of these studies done recently. Emphasis will be placed on increasing the sample size of the database, a few reviews are as follows:

Frid-Adar, Maayan, et al. [12] authors in this paper suggested augmentation scheme based on combination of standard images and synthetic liver lesion generation using GAN in order to improved liver lesion classification. Using GANs, they synthesized high-quality focal liver lesions from CT images and design a CNN for the liver lesion classification task and augmentation of the CNN training set using the generated synthetic data - for improved classification results.

When the authors used classic data augmentation, the performance of the classification yielded 78.6% sensitivity and 88.4% specificity. By adding the synthetic data augmentation, they obtained 85.7% sensitivity and 92.4% specificity.

Iqbal, Talha, and Hazrat Ali. [13] in this work authors propose a new GAN for Medical Imaging (MI-GAN). The MI-GAN generates synthetic retinal image with their segmented masks, which will be used for the application of supervised analysis of medical images. One of the features of the proposed model is the ability to learn useful features from a small training set.

Shmelkov, Konstantin, Cordelia Schmid, and Karteek Alahari. [14] the authors introduce two measures based on image classification GAN-train and GAN-test, which approximate the recall and precision. They suggested new evaluation measures to compare class-conditional GAN architectures with GAN-train and GAN-test scores. they used a neural net architecture for image classification for both these measures. To compute GAN-train, they train a classification network with images generated by a GAN, and then evaluate its performance on a test set composed of real-world images. The authors concluded that generated images are similar to real ones if the classification network, which learns features for discriminating images generated for different classes, can correctly classify real images.

Bowles, Christopher, et al. [15] This paper demonstrates the feasibility of introducing GAN derived synthetic data to the original training datasets in two brain segmentation tasks, which will lead to improve in Dice Similarity Coefficient of between 1 and 5 percentage points under different conditions. They used a Progressive Growing of GANs (PGGAN) network to generate synthetic data. PGGAN was chosen, because the ability of training at large image sizes. synthetic data is added to the re-

al data to increase the size of the dataset and the best result obtained when the authors used GAN + Rotation augmentation.

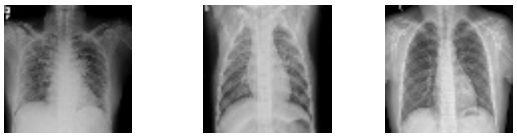
Zeid Baker, Mousa. [16] the researcher in this paper used Generative Adversarial Networks to generate synthetic images similar to the real dataset in order to expand a dataset. they used two types of experiments were achieved, the first is using fine-tune a Deep Convolutional Generative Adversarial Network for a specific dataset, while the second experiment was used to analyze how synthetic data affect the accuracy in a classification task. authors used three types of datasets MNIST, Fashion-MNIST and Flower photos. The authors conclusion that DCGAN leads to an increase the model accuracy depends on the type of the dataset and the data preprocessing plays a big roll on a DCGANs performance, as for almost any ML algorithm.

Wu, Qiufeng, Yiping Chen, and Jun Meng. [17] in this work, authors used Generated images augmented by deep convolutional generative adversarial networks (DCGAN) and original images to Identify Tomato Leaf Disease. They used GoogLeNet classifier to training and testing 5 classes of tomato leaf images, this model achieved accuracy of 94.33%. the authors result that images generated by DCGAN not only enlarge the size of the data set, but also have the characteristics of diversity, which makes the model have a good generalization effect.

## 3. Material and method

### 3.1. Datasets

The dataset was analyzed and preprocessed of three different classes which contain 515 of COVID-19, Pneumonia and Normal images, were selected for this study to aim for the highest variance among classes. where 427 images used for training and 88 images used for validation purpose, which mean 20% of the total images were used for validation. All the images were collected in total from Kaggle dataset in high resolution images. Figure 3 shows samples of these diseases.



(a) (b) (c)

**Figure 3.** X-Ray images (a) COVID-19 (b) Pneumonia (c) Normal

The size of all images should be the same, colored and formatted. In machine learning and deep learning, accuracy of model will be affected if the samples are not equally distributed [18].

**Table 1**

Samples of Chest diseases (X-Ray images) in original dataset

| Diseases | Training | Validation |
|---|---|---|
| COVID-19 | 155 | 33 |
| Pneumonia | 104 | 22 |
| Normal | 168 | 33 |
| Total | 427 | 88 |

In addition, as it shown in Table 1, 155 images of covid-19, 104 images of Pneumonia and 168 images of normal have been used to train our model. There are two issue should be considered: The number of parameters when the number of network layers increases, and the small number of Chest diseases which we have collected. That will lead to the overfit under the influence of many parameters and few datasets. Increasing of the dataset is an effective way to solve this issue. In this paper, authors proposed DCGAN network to increase COVID-19 dataset by generate synthetic dataset. Therefore, the training data set, and validation set will be increased (data enhancement).

**3.2. Model building**

The main goal in this work is to generate images that look like realistic images in each class in order to solve the problem of insufficient medical images. in addition, using GMD model to classify Corona virus with and without DCGAN.

**3.2.1. GMD model**

As shown in Figure 3, the size of input image is set to 150×150 pixels with 3 RGB channels. To extract the features from the image, two convolution layers were used: the first is 256 filters of size 3×3 pixels and the second is 128 filters of size 3×3 pixels. For the pooling layer, a window of size 2×2 pixels were used, which compresses the original image size for further processing. After that, another three convolution layers were used of 128, 64, and 32 filters with size 3×3 pixels with a maximum pooling size 2×2 pixels. Then, fully connection is used (Dense 100 units and 60 units) and output layer (4 units) to predict the Corona virus. CNNs adjust their filter weights through backpropagation, which means that after the forward pass, the network is able to look at the loss function and make a backward pass to update the weights.

The GMD model performance was evaluated against training, testing, and validation datasets using the accuracy measure, which reflects the ability of the model to classify the whole database into four categories with almost no error. The following represents the accuracy measure equation [19]:

$$\text{Accuracy} = \text{TP} / \text{TP} + \text{FP} + \text{FN} + \text{TN}. \quad (4)$$

Where, TP is True Positive: Your predicted positive and it is true, TN is True Negative: Your predicted negative and it is true, FP is False Positive: your predicted positive but it is false and False Negative FN: Your predicted negative but it is false.

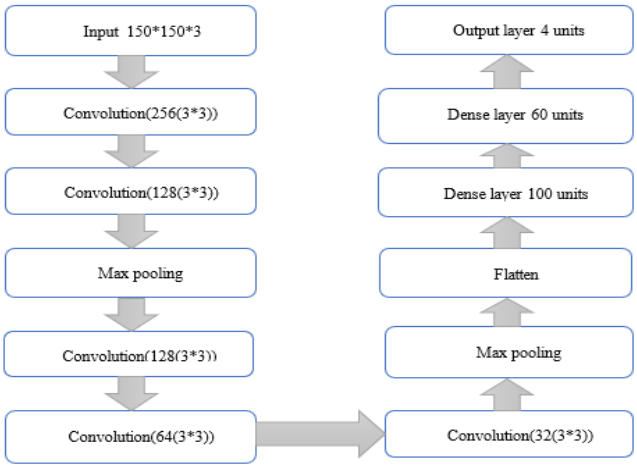



**Figure 3.** Block diagram of GMD model

### 3.2.2. DCGAN architecture

GAN consists of two separate neural networks, Generator (G) and Discriminator (D) constantly competing against each other, the generator, which tries to generate examples that seem the real data, it works to deceive the discriminator, while the role of discriminative network is to distinguish between the real and fake data. a vector of random number will be feeds to the generator, and the outputs are unreal synthetic data denoted by G (z), the discriminator feeds real data and the output of generator. The output of the discriminator is the possibility whether the entered data is real or fake, the output denoted by D (x). The output D (x) represents the probability that x will be a real image if it is 1, which means 100% is a real image, and the output is 0 which means that the image cannot be real [20].

Objective function and how GAN is work, we mentioned above in introduction in section 1. DCGAN is Deep Convolutional GAN, it only replaces the G and D with two convolutional neuronal networks (CNN). Compared with GAN, DCGAN made some changes to the CNN architecture to improve sample quality and convergence speed. these changes are mentioned above in section 1.3.1. In this paper, the block diagram of DCGAN is shown in Figure 4**.** We also will interduce the proposed generator and discriminator network used by referring to the DCGAN architectures as shown in Figure 5.

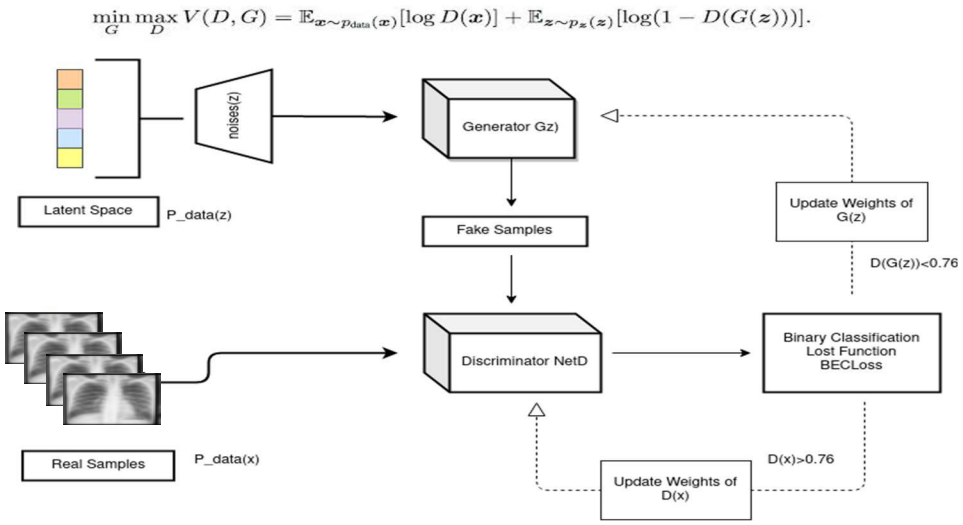



**Figure 4.** Block diagram of DCGAN

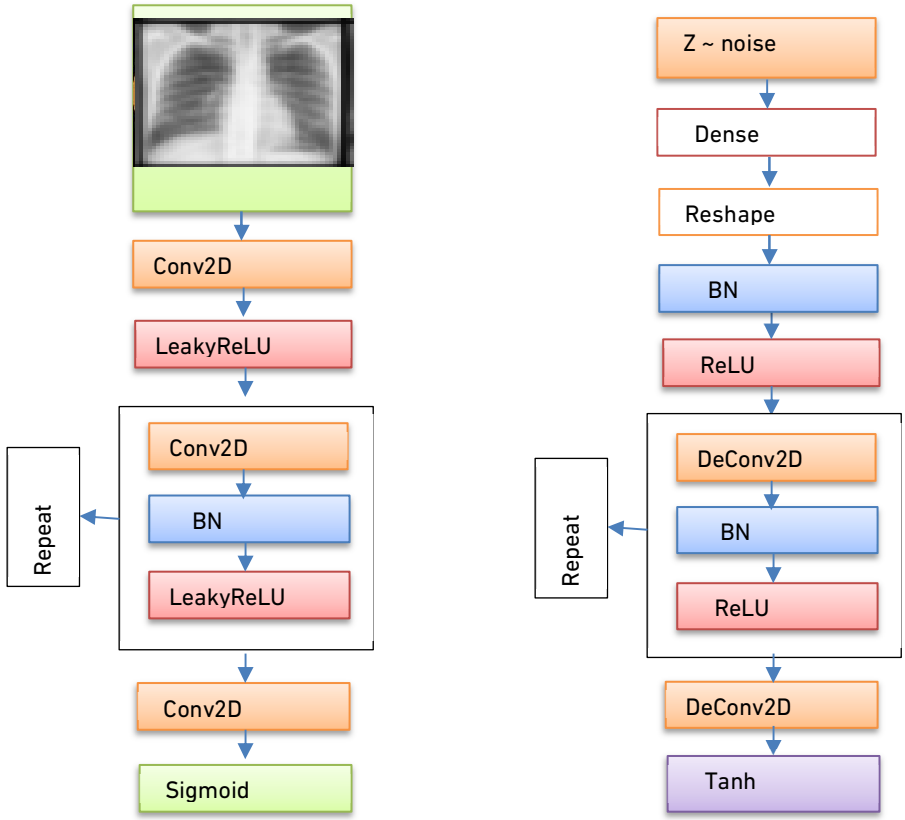



**Figure 5.** discriminator (left) and generator architecture (right)

## 4. Experiments and Results

According to the models explained above, all these models were implemented by Python language (Google Colab) using dual Graphics Processing Unit (GPU). The main objective of this paper is to increase our dataset using DCGAN. These data will be feds to GMD model in order to improve COVID-19 disease classification. Accuracy will be measured before and after adding synthetic dataset.

Dataset has been explained in section 3.1. it consists of three types of Chest diseases, covid-19, Pneumonia and Normal. Table 1 shows the number of samples of each class in training set and validation set.

### 4.1. DCGAN results

DCGAN were used to increase our dataset in order to improve the diagnosis of COVID-19 disease. The schema of our experiment as shown in Figure 4 has been applied. We utilized our dataset which consists of 427 training samples, 88 validation samples of Chest X-Ray Images size 64x64 colored images for three general classes. We used the same network structure for the generator and discriminator as shown in figure 5, and training our medical images, the parameters of the discriminator and generator are updated in the same number of iterations. During this process BCElose will be calculated on all real and fake batch, and calculate gradients for D and G in backward propagation.

| | |
|---|---|
| # (1) Update D network: maximize log(D(x)) + log(1 - D(G(z))) | output = netD(fake.detach().to(device)[:1]).view(-1) |
| ## Train with all-real batch | # Calculate D's loss on the all-fake batch |
| netD.zero_grad() | errD_fake = criterion(output, label) |
| # Format batch | # Calculate the gradients for this batch |
| real_cpu = data[0].to(device)[:1] | errD_fake.backward() |
| b_size = real_cpu.size(0) | D_G_z1 = output.mean().item() |
| label = torch.full((b_size,), real_label, dtype= torch.float, de | # Add the gradients from the all-real and all- fake batches |
| vice=device) | errD = errD_real + errD_fake |
| # Forward pass real batch through D | # Update D |
| output = netD(real_cpu).view(-1) | optimizerD.step() |
| # Calculate loss on all-real batch | # (2) Update G network: maximize log(D(G(z))) |
| errD_real = criterion(output, label) | netG.zero_grad() |
| # Calculate gradients for D in backward pass | la- bel.fill_(real_label) # fake labels are real for gen erator cost |
| errD_real.backward() | # Since we just updated D, perform another forw ard pass of all- fake batch through D |

| | |
|---|---|
| D_x = output.mean().item() | output = netD(fake.to(device)[:1]).view(-1) |
| ## Train with all-fake batch | # Calculate G's loss based on this output |
| # Generate batch of latent vectors | errG = criterion(output, label) |
| noise = torch.randn(b_size, nz, 1, 1, device=device) | # Calculate gradients for G |
| # Generate fake image batch with G | errG.backward() |
| fake = netG(noise) | D_G_z2 = output.mean().item() |
| label.fill_(fake_label) | # Update G |
| # Classify all fake batch with D | optimizerG.step() |

For the discriminant, the label for the true sample is marked with the number 1, and the label for the generated sample is marked as 0, regardless of the quality of the sample generated. So, D(x) should be close to 1 and D(G(z)) should be close to 0. In our experiment we save just the images that D(G(z))>=0.76 which close to 1.

```
x=0.76
if errD_fake >= float(x) or errD_fake == errD_real :
vutils.save_image(vutils.make_grid(fake[:1].to(device)[:1], padding=2, no
rmal-
ize=True), "%s/fake_samples_epoch_%03d.png" % ("/content/drive/MyD
rive/GAN2/FakeImages", epoch), normalize = False)
```

Adam was used as the gradient method for learning parameters of model. Its initial learning rate is 0002.

```
optimizerD = optim.Adam(netD.parameters(), lr=lr, betas=(beta1, 0.999))
optimizerG = optim.Adam(netG.parameters(), lr=lr, betas=(beta1, 0.999))
```

The number of epochs we have used is 4000, that is not mean 4000 images will be generated, because we save the images that D(G(z))>=0.76. Table 2 shows the synthetic images and total images which we obtained. Figure 6 shows samples of the fake and real images.

**Table 2**

Samples of Chest diseases (X-Ray images) using DCGAN

| | Synthetic data using DCGAN | Original data + synthetic data | |
|---|---|---|---|
| Diseases | | Training | Validation |
| COVID-19 | 192 | 347 | 44 |
| Pneumonia | 150 | 254 | 44 |
| Normal | 61 | 229 | 27 |
| Total | 403 | 830 | 246 |

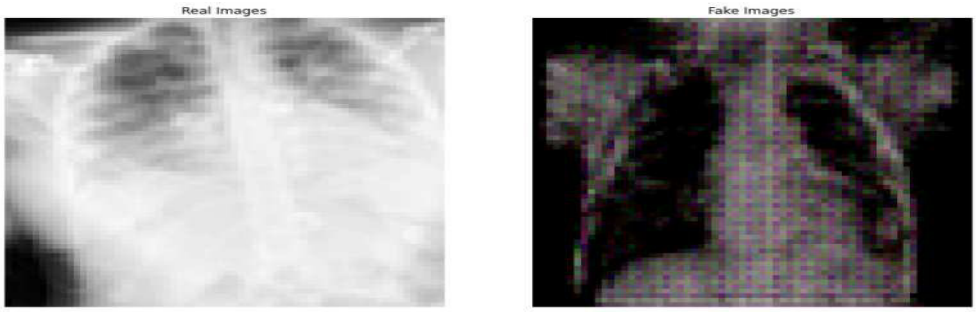



**Figure 6.** Sample of real and fake image

The figures below show the loss during training in both Generator and Discriminator for all classes.

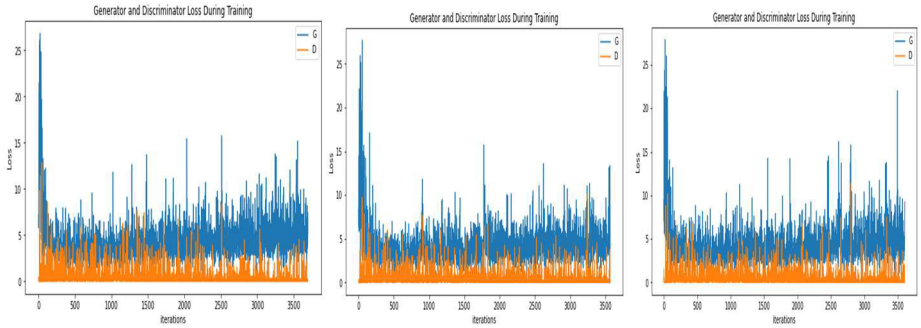



**Figure 7.** Loss in Normal Figure 8: Loss in Covid19 Figure 9: Loss in Pneumonia

### 4.2. GMD model Results with Original dataset

GMD model has been applied, *Constant learning rate* is the default learning rate schedule in SGD. Both momentum and decay rate are set to zero. Learning rate was set to 0.001 which showed a good performance to start. In constant learning rate schedule, the model was defined and the learning rate value was set to constant as follows.

*# define CNN_ model*
*model1 = CNN_model()*

*epochs = 200*
*# define SGD optimizer and set to default except lr*
*learning_rate = 0.001*
*sgd = SGD(lr=learning_rate, momentum=0.0, decay=0.0, nesterov=False).*

Then, compiling, fitting GMD model and plotting the model accuracy as shown in Figure 8. We obtained Accuracy on training set 86,32% and Accuracy on validation set 85,50%.

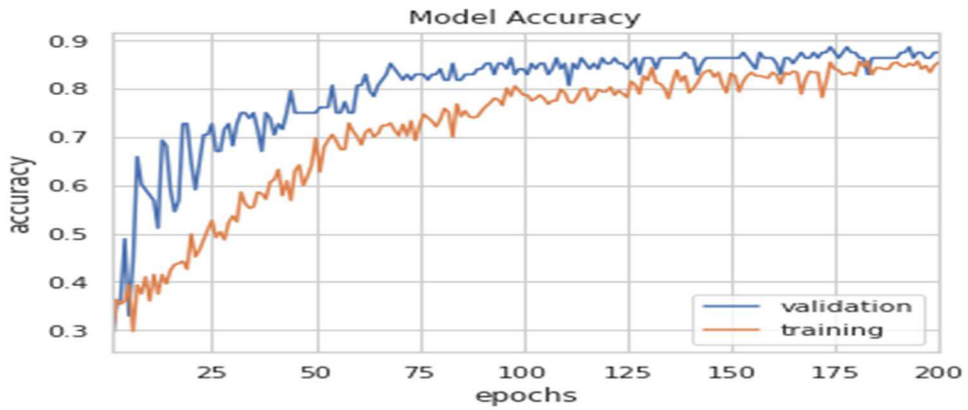



**Figure 8.** Model accuracy using constant learning rate without data generated

### 4.3. GMD model Results with synthetic data

The model has been applied and c*onstant learning rate* is the default learning rate schedule in SGD as mentioned in paragraph 4.1. Then, compiling, fitting our model and plotting the model accuracy as shown in Figure 9. The result obtained, accuracy on training set 95.45% and accuracy on validation set 86.42%.

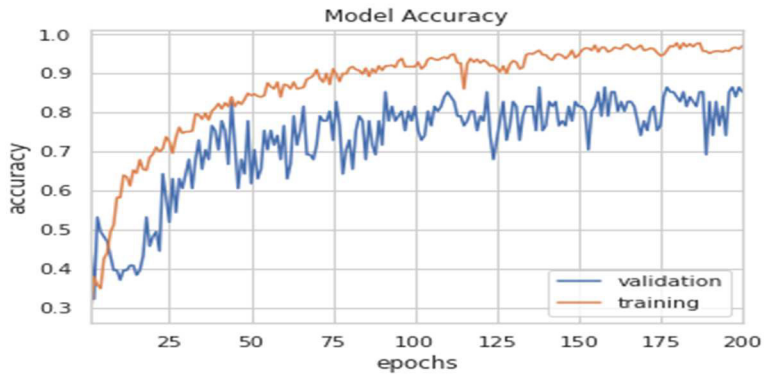



**Figure 9.** Model accuracy using constant learning rate with synthetic data

## 5. Conclusion

In this paper, a Deep Convolutional Generative Adversarial Network, and GMD model based on Keras, TensorFlow and pycharm has been deployed using Python on a dataset of three types of Chest diseases: Covid-19, Pneumonia and Normal.

The implemented DCGAN and GMD model have been trained using GPU in Google Colab, in order to increase our dataset to detect Corona virus.

In conclusion, DCGAN strongly can be used for generating high quality of medical images and increasing training data. Furthermore, the performance of GMD model with original data and with synthetic images data were compared. We confirmed that with the help of DCGAN, GMD model (CNN model) can yield a better *accuracy* compared to traditional methods. Moreover, the *accuracy* of the model had improved significantly from 86.32% in training set and 85.50% in validation set, to 95.45% in training set and 86.42% in validation set. Based on this work, we suggest that DCGAN can be optimized for multi-class generator. We also suggest that this work can be applied to other image classification models such as Vgg16, Inception v3, ResNet to enhance the accuracy.

**1.** Dysregulation of immune response in patients with Covid-19 in Wuhan, China / C. Qin, L. Zhou, Z. Hu, S. Zhang, S. Yang, Y. Tao, C. Xie, K. Ma, K. Shang, W. Wang et al. Clinical Infectious Diseases, 2020. **2.** Who director-general's opening remarks at the media brie_ng on covid-19. 11 march 2020. [Online]. URL: https://www.who.int/dg/speeches/detail/whodirector-general-s-opening-remarks-at-the media-brie_ng-on-covid-19| 11-march-2020 (accessed date: 20.01.2021). **3.** A deep learning system to screen novel coronavirus disease 2019 pneumonia / X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su et al. *Engineering*. **4.** H. Greenspan, B. van Ginneken, and R. M. Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*. May 2016. Vol. 35, no. 5, pp. 1153–1159. **5.** Mustaffa Hussain. GANs-What and Where? 2020. URL: https://medium.com/thecyphy/gans-what-and-where-b377672283c5 (accessed date: 20.01.2021). **6.** Goodfellow Ian J., et al. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*. 2014. **7.** Radford Alec, Luke Metz and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*. 2015. **8.** Mirza Mehdi and Simon Osindero. Conditional generative adversarial nets. 2014. arXiv preprint arXiv:1411.1784. **9.** Mao, Xudong, et al. Least squares generative adversarial networks. *Proceedings of the IEEE international conference on computer vision*. 2017. **10.** Berthelot, David, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative ad-

versarial networks. arXiv preprint arXiv:1703.10717 (2017). **11.** Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. *International Conference on Machine Learning*. ICML, 2017. **12.** Frid-Adar, Maayan, et al. Synthetic data augmentation using GAN for improved liver lesion classification. 2018. *IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018. **13.** Iqbal, Talha, and Hazrat Ali. Generative adversarial network for medical images (MI-GAN). *Journal of medical systems* 42.11 (2018). 1–11. **14.** Shmelkov, Konstantin, Cordelia Schmid, and Karteek Alahari. How good is my GAN? *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018. **15.** Bowles, Christopher, et al. Gan augmentation: Augmenting training data using generative adversarial networks. arXiv preprint arXiv:1810.10863 (2018). **16.** Zeid Baker, Mousa. Generation of Synthetic Images with Generative Adversarial Networks. 2018. **17.** Wu, Qiufeng, Yiping Chen, and Jun Meng. DCGAN-based data augmentation for tomato leaf disease identification. IEEE Access 8 (2020). 98716–98728. **18.** F. J. Pulgar, A. J. Rivera, F. Charte, and M. J. del Jesus. On the impact of imbalanced data in convolutional neural networks performance. *In Proc. Int. Conf. Hybrid Artif. Intell. Syst*. Vol. 10334. Cham, Switzerland : Springer, 2017. Pp. 220–232. **19.** Visa, Sofia, et al. Confusion Matrix-based Feature Selection. MAICS 710 (2011). 120–127. **20.** M. Mirza and S. Osindero. Conditional generative adversarial nets. 2014, arXiv:1411.1784. [Online]. URL: http://arxiv.org/abs/ 1411.1784 (accessed date: 20.01.2021).

## REFERENCES:

**1.** Dysregulation of immune response in patients with Covid-19 in Wuhan, China / C. Qin, L. Zhou, Z. Hu, S. Zhang, S. Yang, Y. Tao, C. Xie, K. Ma, K. Shang, W. Wang et al. Clinical Infectious Diseases, 2020. **2.** Who director-general's opening remarks at the media brie_ng on covid-19. 11 march 2020. [Online]. URL: https://www.who.int/dg/speeches/detail/whodirector-general-s-opening-remarks-at-the media-brie_ng-on-covid-19| 11-march-2020 (accessed date: 20.01.2021). **3.** A deep learning system to screen novel coronavirus disease 2019 pneumonia / X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su et al. *Engineering*. **4.** H. Greenspan, B. van Ginneken, and R. M. Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*. May 2016. Vol. 35, no. 5, pp. 1153–1159. **5.** Mustaffa Hussain. GANs-What and Where? 2020. URL: https://medium.com/thecyphy/gans-what-and-where-b377672283c5 (accessed date: 20.01.2021). **6.** Goodfellow, Ian J., et al. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*. 2014. **7.** Radford Alec, Luke Metz and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*. 2015. **8.** Mirza Mehdi and Simon Osindero. Conditional

generative adversarial nets. 2014. arXiv preprint arXiv:1411.1784. **9.** Mao, Xudong, et al. Least squares generative adversarial networks. *Proceedings of the IEEE international conference on computer vision*. 2017. **10.** Berthelot, David, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717 (2017). **11.** Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. *International Conference on Machine Learning*. ICML, 2017. **12.** Frid-Adar, Maayan, et al. Synthetic data augmentation using GAN for improved liver lesion classification. 2018. *IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018. **13.** Iqbal, Talha, and Hazrat Ali. Generative adversarial network for medical images (MI-GAN). *Journal of medical systems* 42.11 (2018). 1–11. **14.** Shmelkov, Konstantin, Cordelia Schmid, and Karteek Alahari. How good is my GAN? *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018. **15.** Bowles, Christopher, et al. Gan augmentation: Augmenting training data using generative adversarial networks. arXiv preprint arXiv:1810.10863 (2018). **16.** Zeid Baker, Mousa. Generation of Synthetic Images with Generative Adversarial Networks. 2018. **17.** Wu, Qiufeng, Yiping Chen, and Jun Meng. DCGAN-based data augmentation for tomato leaf disease identification. IEEE Access 8 (2020). 98716–98728. **18.** F. J. Pulgar, A. J. Rivera, F. Charte, and M. J. del Jesus. On the impact of imbalanced data in convolutional neural networks performance. *In Proc. Int. Conf. Hybrid Artif. Intell. Syst*. Vol. 10334. Cham, Switzerland : Springer, 2017. Pp. 220–232. **19.** Visa, Sofia, et al. Confusion Matrix-based Feature Selection. MAICS 710 (2011). 120–127. **20.** M. Mirza and S. Osindero. Conditional generative adversarial nets. 2014, arXiv:1411.1784. [Online]. URL: http://arxiv.org/abs/ 1411.1784 (accessed date: 20.01.2021).

---

**Ярощак С. В., к.т.н., доцент, Смайда Махмуд** (Національний університет водного господарства та природокористування, м. Рівне)

## СТВОРЕННЯ РЕАЛІСТИЧНИХ МЕДИЧНИХ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ DCGAN ДЛЯ ПОКРАЩЕННЯ КЛАСИФІКАЦІЇ COVID-19

**При класифікації зображень найважливішими факторами є обсяг даних, особливо у випадку медичних зображень. Однак отримання медичних зображень є великою проблемою. У цій роботі ми створили глибоку згорткову генеративно змагальну мережу (DCGAN), яка генерує синтетичні медичні зображення. Крім того, для ідентифікації коронавірусу будемо використовувати нашу модель, яку розробили для класифікації захворювань очей. Щоб виміряти точність моделі, протестуємо її на згенерованих синтетичних медичних зображеннях та без них. В якості набору даних будуть використані**

**рентгенівські знімки органів грудної клітини трьох типів: Covid-19, пневмонія та нормальний стан. Кількість знімків кожного типу, відповідно, рівна 155, 104 та 168. На основі цих знімків та з використанням DCGAN було згенеровано синтетичні зображення та класифіковано їх GMD методом. Після чого навчено нашу модель на всьому наборі даних, реальних зображеннях разом з синтетичними, та порівняно результати.**

**Точність моделі значно покращилася з 86,32% на навчальному наборі та 85,50% на наборі перевірки, до 95,45% на навчальному наборі та 86,42% на наборі перевірки. В зв'язку з цим, підхід з розширенням вхідних даних синтетичними зображеннями доцільно застосовувати і з іншими моделями класифікації зображень, такими як Vgg16, Inception v3, ResNet для підвищення їх точності. Варто відзначити, що кількість епох є дуже важливою при створення високоякісних зображень.**

***Ключові слова*: Covid19; глибинне навчання; DCGAN; медичні зображення.**

---

**Ярощак С. В., к.т.н., доцент, Смайда Махмуд** (Национальный университет водного хозяйства и природопользования, г. Ровно)

## СОЗДАНИЕ РЕАЛИСТИЧНЫХ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ DCGAN ДЛЯ УЛУЧШЕНИЯ КЛАССИФИКАЦИИ COVID-19

**При классификации изображений важнейшими факторами являются объем данных, особенно в случае медицинских изображений. Однако получение медицинских изображений является большой проблемой. В этой работе мы создали глубокую сверточную генеративно состязательную сеть (DCGAN), которая генерирует синтетические медицинские изображения. Кроме того, для идентификации коронавируса будем использовать нашу модель, разработанную для классификации заболеваний глаз. Чтобы измерить точность модели, протестируем ее на сгенерированных синтетических медицинских изображениях и без них. В качестве набора данных будут использованы рентгеновские снимки органов грудной клетки трех типов: Covid-19, пневмония и нормальное состояние. Количество снимков каждого типа, соответственно, равна 155, 104 и 168. На основе этих снимков и с использованием DCGAN было сгенерированы синтетические изображения и классифицированы**

**их GMD методом. После чего на всем наборе данных, реальные изображения вместе с синтетическими, обучено нашу модель, и сравнено результаты.**
**Точность модели значительно улучшилась с 86,32% на учебном наборе и 85,50% на наборе проверки, до 95,45% на учебном наборе и 86,42% на наборе проверки. В связи с этим, подход с расширением входных данных синтетическими изображениями целесообразно применять и с другими моделями классификации изображений, такими как Vgg16, Inception v3, ResNet для повышения их точности. Стоит отметить, что количество эпох очень важно при создании высококачественных изображений.**
***Ключевые слова*: Covid19; глубокое обучение; DCGAN; медицинские изображения.**

---