Medical X-ray Image Classification Employing DCGAN and CNN Transfer Learning Techniques

Md. Asif Talukdar[1,2], Ayesha Siddika[1,3] Ahasanul Haque Abir[1,4], Mohammed Ziad Hassan[1,5], and Muhammad Iqbal Hossain[1,6]

[1] BRAC University, Dhaka, Bangladesh
[2] md.asif.talukdar@g.bracu.ac.bd
[3] ayesha.siddika2@g.bracu.ac.bd
[4] ahasanul.haque.abir@g.bracu.ac.bd
[5] mohammed.ziad.hassan@g.bracu.ac.bd
[6] iqbal.hossain@bracu.ac.bd

Abstract. Over the decades, a typical imaging test that has been used is an X-ray. It allows doctors to see into the body without an incision. As a result, an X-ray can aid in diagnosing, monitoring, and treating a variety of medical disorders by detecting diseases beforehand. Among the diseases, pneumonia got major heed because of its intensity. As the lungs are the most vulnerable part of the body when it comes to pneumonia, doctors rely on the chest X-Ray to diagnose the disease. In this research, we have worked on the X-ray images to discern pneumonia using our custom CNN model and different types of Transfer Learning models and manifested a comparison of those methods in terms of their ability to detect the disease. Furthermore, we performed Generative adversarial networks (GAN) with deep convolutional layers to generate and merge a new training dataset using existing image data. Then, we executed the models anew after acquiring a new artificial dataset. Before using GAN we got accuracy of 94%, 94%, 73%, 73%, 96%, 97% and 94% in Custom CNN, InceptionV3, ResNet50, EfficientNetB0, VGG16, DenseNet201 and Xception respectively. However, we observed improved accuracy from all models applying GAN except for DenseNet201. Moreover, VGG16, DenseNet201, and Custom CNN acquired the higher accuracy overall.

Key words: DCGAN, Transfer Learning, X-ray, Pneumonia, Synthetic Image & Convolutional Neural Network.

1 Introduction

Pneumonia is a respiratory infection in which the air sacs in the lungs become inflamed. Coughing, fever, chills, and trouble breathing can occur when the air sacs become blocked with fluid or pus. A complication such as an empyema or the growth of an abscess could result in a lack of response. Doctors usually perform a physical exam and a chest x-ray to examine the lungs, heart, and blood vessels to diagnose pneumonia. When reading the x-ray, the radiologist will look for white areas in the lungs that indicate an infection. Other pneumonia-related disorders, such as abscesses or pleural effusions, will be visible on this X-ray image [1].

Medical imaging findings can be processed more quickly with computer-aided detection and diagnosis using machine learning techniques. Convolutional Neural Networks (CNN), one of the most well-known machine learning algorithms for image categorization, will be used to achieve our goal. This technique enables machines to predict and successfully label new images [2]. Furthermore, this method has been shown to extract beneficial characteristics from images in image classification applications [3]. There have been numerous CNN architectures produced to date. In this study, we attempt to demonstrate and compare several of these CNN architectures so that clinicians may make better decisions when diagnosing pneumonia. We utilized DCGAN to pit two neural networks against one other to generate new, synthetic data samples that may pass for accurate data to achieve improved outcomes. Our ultimate goal is to develop a process that would allow doctors to diagnose a condition quickly.

2 Literature Review

In the last few years, the application of deep learning (DL) has grown exponentially in the healthcare sector, and various researches prove that DL models can be used for the detection of different diseases using image classification. Some of those are: In comparing two CNN networks Xception and VGG16, Ayan et al. [4] showed that vgg16 performs better for detecting the typical case, and on the other hand, Xception performs better for the detection of pneumonia, and the combination of both will be more accurate. However, on a profound scale, Deep learning algorithms were utilized by Srivastav et al. to categorize chest X-ray images to diagnose pneumonia [5]. First, deep convolutional generative adversarial networks(DCGAN) were trained to supplement synthetic images and oversample the dataset to improve the model's performance. Then, using VGG16 as the foundation model for image classification, transfer learning was applied with convolutional neural networks(CNN). On the validation set, the model had a 94.5% accuracy rate. Furthermore, the accuracy of the proposed model was found to be significant compared to the naive models. Lastly, Rodŕıguez et al. used a neuroevolution algorithm based on Particle Swarm Optimization for the construction and training of GANs to develop biomedical Chest X-Ray (CXR) pictures of pneumonia caused by COVID-19 [6]. The suggested method allows for creating a swarm of GAN topologies, each of which grows incrementally while being trained at the same time. For the synthesis of CXR pictures, the suggested approach achieves better FID outcomes than handcrafted GANs.

3 Methodology

Our study explores the usability and application of Deep Convolutional Generative Adversarial Network (DCGAN) in detecting pneumonia from X-Ray images. Generative Adversarial Networks is unsupervised learning using the deep learning method, which looks for patterns in input data to generate new synthetic output [7]. We used DCGAN to generate new X-Ray data using the existing dataset. Firstly, we used our original dataset in our models and later used the original dataset with DCGAN generated data to improve the final result.

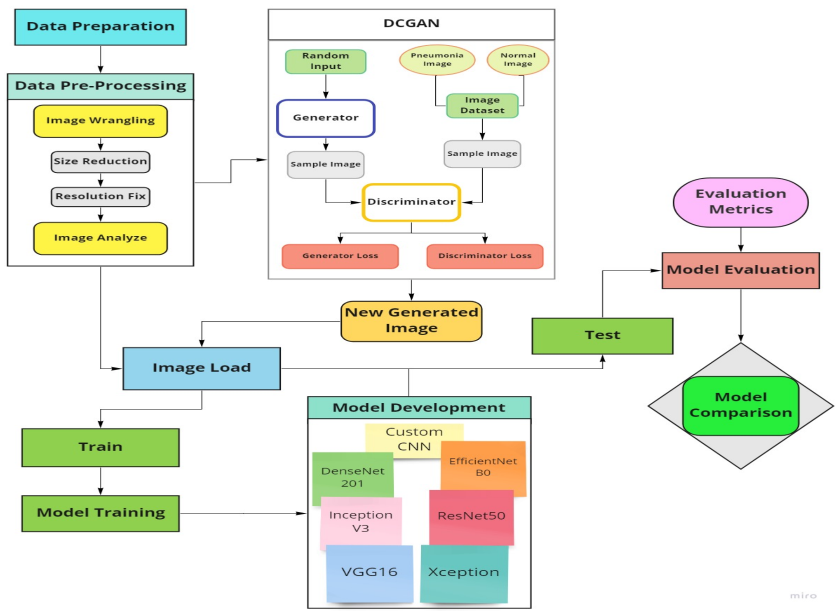

Fig. 1. Workflow Diagram with DCGAN Architecture

3.1 Data Acquisition and Pre-Processing

We used an X-Ray dataset for our study from a subsidiary of Google called Kaggle [8]. Our dataset contains Chest X-Ray Images, which include Normal Chest X-rays and Pneumonia Chest X-rays. Dataset was divided and stratified into a training set containing 80 percent of data and a test set containing 20 percent of data. Our dataset contains a total of 5856 images (73% pneumonia and 27% normal). All the images were resized to 128x128 pixels. For the DCGAN model, we used training dataset of X-ray images to generate synthetic image dataset of 64x64 pixels which were upscaled to 128x128 pixels for optimal result. We applied 150-epoch approach to generate 1500 synthetic pneumonia and 100-epoch approach to generate 1000 synthetic normal image data using DCGAN model. For both of the cases we selected approximately the last 7% of the newly generated synthetic images.

3.2 Model Selection

Deep Convolutional Generative Adversarial Network (DCGAN) DCGAN is a process with Deep Convolutional layers that can create synthetic images referencing authentic images that look like an actual image. DCGAN consists of two models, Generator, also known as Artist, and Discriminator, also known as the Critic [9]. In DCGAN, the Artist Generator creates images using random data. Then, the Discriminator gets the synthetic image created by Generator and takes sample images from the actual images dataset for reference. During the process it produces Generator loss and Discriminator loss which will determine whether the generated synthetic image is acceptable or not. For standard loss function, if Generator loss is decreasing and Discriminator loss is increasing, that means the Generator is moving towards making a synthetic image that looks like an actual image [10]. The Generator will keep generating images using random data, and after each approach, it will try to improve its creation from the previous one based on the output of the loss function.

Custom Convolutional Neural Network Convolutional Neural Networks (CNN) are commonly used to analyze visual imagery. CNN uses a unique Convolution technique rather than traditional Matrix Multiplication. The setup of our Custom Convolutional Neural Network is given in Table 1.

CNN Transfer Learning Techniques Our study includes Transfer Learning techniques such as InceptionV3, ResNet50, EfficientNetB0, VGG16, DenseNet201 and Xception.

4 RESULT AND ANALYSIS

4.1 Experimental Setup:

The Deep learning CNN models, including DCGAN and learning models, were trained on a laptop using a Google Colab environment using GPU and high ram configuration. Libraries used within this work include Pandas, Numpy, Seaborn, Matplotlib, Scikit-learn, Keras, PIL, and Tensorflow.

4.2 Model Description

Model setups for our DCGAN and Deep Convolutional Neural Network (DCNN) are given in table 1. All the models are trained with Loss function Binary Cross Entropy and Learning Rate was set to 0.0001. All the models in our work is optimized by Adam optimizer except for InceptionV3 and ResNet50 using RMSprop and SGD as optimizer respectively.

Table 1. DCGAN and DCNN model setup

Techniques	DCGAN		Deep Convolutional Neural Network	
Models	**Generator**	**Discriminator**	**Custom CNN**	**Transfer Learning CNN**
Model Structure	Input (100) Dense_1 (8192) Reshape (8,8,128) Conv2DT filter = 128, kernel = 4, stride = 2 Leaky Relu (alpha = 0.2) Conv2DT filter = 256, kernel = 4, stride = 2 Leaky Relu (alpha = 0.2) Conv2DT filter = 512, kernel = 4, stride = 2 Leaky Relu (alpha = 0.2) Conv2D filter = 3, kernel = 5	Input (64,64,3) Conv2D filter = 64, kernel = 4, stride = 2 Leaky Relu (alpha = 0.2) Conv2D filter = 128, kernel = 4, stride = 2 LeakyRelu (alpha = 0.2) Conv2D filter = 128, kernel = 4, stride = 2 Leaky Relu (alpha = 0.2) Flatten Dropout (0.2) Dense_1	Conv2D filter = 32 kernel = 3 Conv2D filter = 64 kernel = 3 MaxPool2D = 2 Conv2D filter = 128 kernel = 3 MaxPool2D = 2 Conv2D filter = 256 kernel = 3 MaxPool2D = 3 Flatten Dense_120 Dense_120 Dense_60 Dropout (0.2) Dense_1	Base Model output layer trainable = False layers Flatten Dense_1024 Dense_512 Dropout = 0.2
Padding		same		valid
Number of epochs	100-150		20	20
Activities	Leaky Relu, sigmoid		Relu, sigmoid	Relu, sigmoid
Parameters (Trainable)	3,750,275	404,801	1,177,801	

4.3 Comparison of Custom CNN

Table 2 and Table 3 illustrates the accuracy, precision, recall, and f1 score of our custom CNN and other six models before and after applying DCGAN respectively. Before using DCGAN, our custom CNN obtained 0.92 Precision, 0.93 Recall, and an F1 score of 0.92. Our custom CNN performed better when we utilized DCGAN, with 0.95 Precision, 0.95 Recall, and 0.95 F1 scores. Before utilizing DCGAN, we had a test accuracy of 94.03 percent, and after using DCGAN, we had a test accuracy of 96.09 percent. For both cases, The Custom CNN model was trained for 20 epochs.

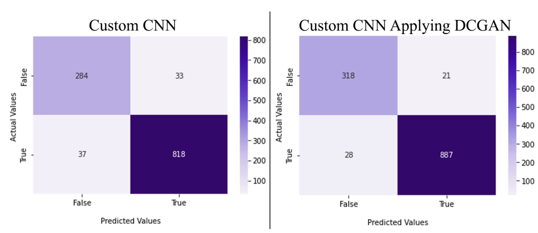

Fig. 2. Custom CNN confusion matrix

Furthermore, as shown in the Heatmap diagram, our custom CNN correctly recognized 818 cases as pneumonia before applying DCGAN and 887 cases as pneumonia after applying DCGAN. For normal cases, the model correctly recognized 284 cases before DCGAN was applied and accurately recognized 318 cases after applying DCGAN on the dataset. With regards to the classification of pneumonia and normal X-Ray images, it can be observed that our custom CNN provided significantly better performance when DCGAN was employed.

4.4 Comparison of Transfer learning models

Using Transfer Learning techniques for Convolutional neural networks, we examined six different networks: Inception v3, RestNet50, EfficientNetB0, VGG16, DenseNet201, and Xception before and after using DCGAN. All the models were trained for 20 epochs.

Before applying DCGAN, InceptionV3 scored 0.92 on Precision, Recall, and F1 score. It also successfully detected 822 pneumonia cases and 276 normal cases without DCGAN. RestNet50 and EfficientNetB0 performed 0.36 Precision, 0.50 Recall, and a F1 score of 0.42. Moreover, both the models detected 855 pneumonia cases successfully. Furthermore, VGG16 performed 0.96 Precision, 0.94 Recall, and a F1 score of 0.95, and it detected correctly 839 pneumonia and 288 normal cases. Xception performed 0.93 Precision, 0.92 Recall, and a F1 score of 0.92, and it precisely detected 821 pneumonia cases and 281 cases of normal instances. However, DenseNet201 achieved the higest performance with 0.97 Precision, 0.95 Recall, 0.96 F1 score and a accuracy of 96.84 %.

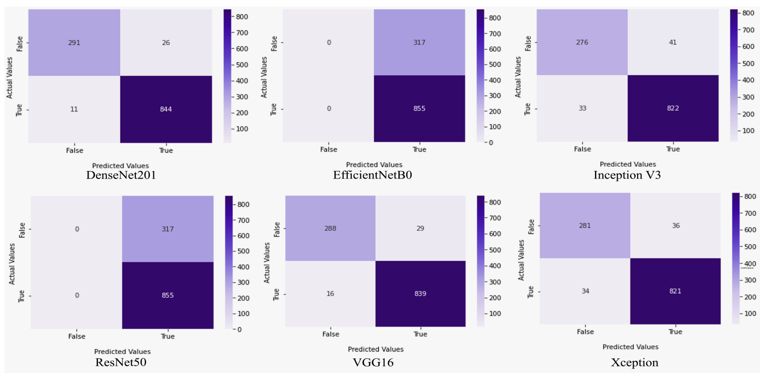

Fig. 3. Transfer Learning confusion matrix before DCGAN applied

Table 2. Comparison of CNN models (without DCGAN)

Model	Precision	Recall	F1 Score	Train Accuracy(%)	Test Accuracy(%)	Loss	Train Time (s)
Custom CNN	0.92	0.93	0.92	99.15	94.03	0.2923	160
Inception V3	0.92	0.92	0.92	99.22	93.69	**0.9448**	129
Resnet50	0.36	0.50	0.42	73.90	72.95	0.5270	133
EfficientNetB0	0.36	0.50	0.42	73.03	72.95	0.5847	129
VGG16	0.96	0.94	0.95	**1.00**	96.16	0.2279	93
DenseNet201	**0.97**	**0.95**	**0.96**	1.00	**96.84**	0.1961	**258**
Xception	0.93	0.92	0.92	99.74	94.03	0.2504	132

Furthermore, after applying DCGAN, we got better performance for almost every transfer learning model. The result shows InceptionV3 scored 0.95 Precision, 0.94 Recall, and an F1 score of 0.94. It also successfully detected 891 pneumonia cases and 307 normal cases after applying DCGAN. However, slight improvement was demonstrated by RestNet50 and EfficientNetB0 in terms of test-accuracy and detected 915 pneumonia cases correctly. Moreover, DenseNet201 and VGG16 performed 0.96 on Precision, Recall, and F1 score. VGG16 detected 897 pneumonia cases and 318 normal cases accurately whereas DenseNet201 detected 899 pneumonia cases and 315 normal cases correctly. Then lastly, Xception performed 0.92 Precision, 0.94 Recall, and an F1 score of 0.93, and it precisely detected 865 pneumonia cases and 317 normal cases after applying DCGAN.

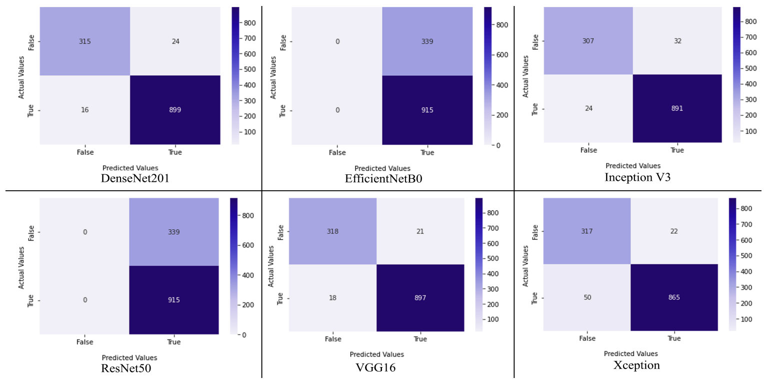

Fig. 4. Transfer Learning confusion matrix after DCGAN applied

Table 3. Comparison of CNN models (DCGAN applied)

Model	Precision	Recall	F1 Score	Train Accuracy(%)	Test Accuracy(%)	Loss	Train Time (s)
Custom CNN	0.95	0.95	0.95	99.23	96.09	0.1865	179
Inception V3	0.95	0.94	0.94	99.63	95.53	**0.7313**	163
Resnet50	0.36	0.50	0.42	75.05	72.97	0.5249	182
EfficientNetB0	0.36	0.50	0.42	74.21	72.97	0.5845	144
VGG16	**0.96**	**0.96**	**0.96**	**1.00**	**96.89**	0.1838	152
DenseNet201	0.96	0.96	0.96	1.00	96.81	0.1563	**328**
Xception	0.92	0.94	0.93	98.88	94.26	0.2593	209

While DenseNet201 had the best accuracy of all the models before employing DCGAN, VGG16 achieved highest accuracy after employing DCGAN. Except for DenseNet201, we observed that using DCGAN improved accuracy for other models. InceptionV3 and VGG16 models accuracy was increased 1.84% and 0.73% after DCGAN applied while our Custom CNN showed a promising accuracy increase of 2.06%. Moreover, From the confusion matrix Heat-Maps, we can conclude that all CNN learning models performed better at detecting real situations applying DCGAN.

5 Conclusion and Future Works

Our purpose in this study was to produce analytical results that would allow us to compare CNN models after employing DCGAN. According to the results of the studies, custom CNN, VGG16 and DenseNet201 are the most accurate of the all methods. Not only that, but we also acquired a remarkable accuracy increase in the VGG16 and Custom CNN approach after fabricating our data set with DCGAN. Therefore, this method can successfully achieve the ultimate goal of obtaining more accurate results in the X-Ray image classification.

However, due to a lack of computational capacity, we might not have achieved the peak results produced by DCGAN. Therefore, in the future, we would like to collect a more extensive data set with the most computational power possible to produce the most remarkable results. In addition, in future work, we will experiment with various medical data images that are not limited to chest X-rays. As a result, the overall classification of medical images will be more effective.

References

1. (ACR) R (2021) Pneumonia. In: Radiologyinfo.org. https://www.radiologyinfo.org/en/info/pneumonia . Accessed 15 December 2021
2. Erickson B, Korfiatis P, Akkus Z, Kline T (2017) Machine Learning for Medical Imaging. RadioGraphics 37:505-515. doi: 10.1148/rg.2017160130
3. (2021) CNN For Image Classification — Image Classification Using CNN. In: Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/ . Accessed 15 December 2021
4. Ayan E, Unver H (2019) Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning. 2019 Scientific Meeting on Electrical-Electronics Biomedical Engineering and Computer Science (EBBT). doi: 10.1109/ebbt.2019.8741582
5. Srivastav D, Bajpai A, Srivastava P (2021) Improved Classification for Pneumonia Detection using Transfer Learning with GAN based Synthetic Image Augmentation. 2021 11th International Conference on Cloud Computing, Data Science Engineering (Confluence). doi: 10.1109/confluence51648.2021.9377062
6. Rodriguez-de-la-Cruz J, Acosta-Mesa H, Mezura-Montes E (2021) Evolution of Generative Adversarial Networks Using PSO for Synthesis of COVID-19 Chest X-ray Images. 2021 IEEE Congress on Evolutionary Computation (CEC). doi: 10.1109/cec45853.2021.9504743
7. Brownlee J (2021) A Gentle Introduction to Generative Adversarial Networks (GANs). In: Machine Learning Mastery. https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/ . Accessed 15 December 2021
8. Kaggle.com. 2021. Chest X-Ray Images (Pneumonia). [online] Available at: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia/ . Accessed 15 December 2021
9. (2021) Deep Convolutional Generative Adversarial Network—TensorFlow Core. In: TensorFlow. https://www.tensorflow.org/tutorials/generative/dcgan . Accessed 15 December 2021
10. Brownlee, J., 2021. A Gentle Introduction to Generative Adversarial Network Loss Functions. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/generative-adversarial-network-loss-functions . Accessed 15 December 2021.