International Journal of Image and Graphics
2350002 (21 pages)
© World Scientific Publishing Company
DOI: 10.1142/S021946782350002X

World Scientific
www.worldscientific.com

Int. J. Image Grap. Downloaded from www.worldscientific.com

Performance Evaluation of Convolutional Neural Network Using Synthetic Medical Data Augmentation Generated by GAN

Ramesh Adhikari*
Pokhara University
Kathmandu, Nepal
ramesh.182980@ncit.edu.np

Suresh Pokharel
Michigan Technological University
Michigan, USA
sureshp@mtu.edu

Received 20 March 2021
Accepted 10 September 2021
Published 28 December 2021

Data augmentation is widely used in image processing and pattern recognition problems in order to increase the richness in diversity of available data. It is commonly used to improve the classification accuracy of images when the available datasets are limited. Deep learning approaches have demonstrated an immense breakthrough in medical diagnostics over the last decade. A significant amount of datasets are needed for the effective training of deep neural networks. The appropriate use of data augmentation techniques prevents the model from over-fitting and thus increases the generalization capability of the network while testing afterward on unseen data. However, it remains a huge challenge to obtain such a large dataset from rare diseases in the medical field. This study presents the synthetic data augmentation technique using Generative Adversarial Networks to evaluate the generalization capability of neural networks using existing data more effectively. In this research, the convolutional neural network (CNN) model is used to classify the X-ray images of the human chest in both normal and pneumonia conditions; then, the synthetic images of the X-ray from the available dataset are generated by using the deep convolutional generative adversarial network (DCGAN) model. Finally, the CNN model is trained again with the original dataset and augmented data generated using the DCGAN model. The classification performance of the CNN model is improved by 3.2% when the augmented data were used along with the originally available dataset.

Keywords: Convolutional neural network; synthetic data augmentation; deep convolutional generative adversarial network (DCGAN); chest X-ray images.

*Corresponding author.

1. Introduction

One of the vital techniques for identifying abnormalities in the medical field is imaging, which relies on image acquisition and analysis. In recent years, to enhance image perception, deep learning models have been commonly utilized. As mentioned by Verma and Prakash,[1] a very significant application of the machine learning approach is deep learning for image classification. One of the biggest challenges today in the application of machine learning and deep learning models for medical image classification is coping with the limited available data. The problem of having a small amount of data in the medical imaging domain has prompted this study to find out effective methods for synthetic data augmentation so that we can enrich the available dataset. Data augmentation artificially expands the size of the training dataset by using computer algorithms. In addition to the traditional/geometric image transformation techniques, synthetic data generated using generative models seem to have more variability and enrich the size of the dataset, as a result of which the classification performance can be improved.[2]

This research mainly focuses on enhancing the performance of the Convolutional Neural Network (CNN) model to classify the normal and pneumonia X-ray images by using synthetic images generated by Generative Adversarial Network (GAN). Although most of the medical databases are publicly accessible, however, some of them are still domain constrained, being restricted for a specific size or merely relevant to specific medical issues. Gathering the onsite natural medical data is a complicated and costly process that comprises the necessity of researchers and radiologists to collaborate in next extent.[11]

Among other deep learning methods, CNN has proven its exemplary performance in different computer vision applications.[5] The CNN consists of webbed layers along with dense layer, max-pooling layer, batch normalization layer, etc.[19] The activation function, ReLU (Rectified Linear Unit), that we used in this work, ensures the nonlinearity in the model helping the network to learn complex patterns in the data fed into it.[7] The model created using a small amount of training data performs poorly when tested with unseen data. Thus, it cannot be generalized especially in the medical field where any kind of analysis is highly critical. Therefore, it is very important to have a substantial amount of data for training the CNN model. In the situation where generating new real data is impossible or costly or time consuming, the synthetic models like GAN can be used to generate more data to overcome the above-mentioned problem.

The prime objective of this research is to enhance the performance of a CNN by using GAN. To achieve this objective, the following are the sub-objectives of this research:

(1) To train the CNN model with chest X-ray images and calculate the classification performance.
(2) To generate the synthetic chest X-ray images using GAN.

Int. J. Image Grap. Downloaded from www.worldscientific.com

(3) To train the CNN model by enriching the existing dataset with synthetic images and compare the classification performance.

2. Literature Review

In 2012, the CNN came into existence with AlexNet.[1] For all image-related topics, CNN is mostly used in the field of classification without any human intervention. For the high accuracy and efficiency of the CNN model, a large amount of the dataset is needed. The conventional methods can be used for data augmentation to enrich the dataset, such as shifting the data, padding, cropping, flipping and rotating of data.[1] Although these techniques improve the training data for machine learning models and are widely used, the images generated do not contain much variability. Therefore, there is a necessity to develop synthetic data that improves the variability resulting in the enriching of the dataset significantly.

GANs introduced by Goodfellow[4] were initially used to generate visually realistic images. A next variant of GANs is called Deep Convolutional GAN (DCGAN),[3] which consists of convolutional layers and uses methods like convolutional stride and transposed convolutions for downsampling and upsampling, respectively. By using DCGAN, it is possible to create synthetic images that are close to real images and they can be used to train the CNN model to improve the classification accuracy.[10] A small dataset of computed tomography (CT) images of 182 liver lesions demonstrates this novel process.[2] The deep learning framework has shown domineering performance on many machine learning problems. CNNs[5] have been most extensively studied among all other alternatives. CNN consists of various types of layers along with the max-pooling layer. It uses the activation function ReLU, which helps to ensure the nonlinearity of the network model.[7]

The recent research has shown that using CNN had enhanced the efficiency of the research, the study of particle collisions in energy physics, which resulted in fantastic outcomes.[8] Deep learning methods have recently shown tremendous potential in image processing and computer vision domain. These methods have been used in various high-performance medical imaging modalities in the segmentation, identification and classification of medical images.[9] Image segmentation methods are used in many medical applications, the new optimal approach for automatic medical image region segmentation has been introduced by Khan and Debnath.[27] In another work, Jayaraman *et al.*[28] proposed a modified flower pollination-based method for performing multilevel segmentation of medical images. This technique explores optimal thresholds in the problem space of the given medical image.

The CNN architecture was influenced by the human brain's Visual Cortex organization.[13] It is essentially a model of deep learning that takes images as inputs, assigns appropriate weights for differentiation from each other and performs any given task, such as the classification with learned inference. Because of their improved performance in image classification, CNNs are very common. From an image, the convolutional layers in the network, along with filters, help to extract the spatial

Int. J. Image Grap. Downloaded from www.worldscientific.com

and temporal characteristics. CNN is essentially a feed-forwarded artificial neural network in terms of architecture (ANNs).[14]

There are many researchers[22–25] who have begun to research on the medical imaging by using GANs, such as in image super-resolution,[22] anomaly detection,[23] estimating the CT images from the corresponding MR images[24] and old image decreasing.[25] The anomalies were found in the medical images by training a generative model, which is based on the deep GANs, and a discriminator network at the same time.[22] By using the Auto-Context Model, the output is increased as the GAN context is effectively extended during the training phase.[24] The generative models are being popular in creating artificial instances from a dataset such that they retain similar characteristics to the original set.[26]

After training the neural network model, it is important to measure its testing accuracy. The confusion matrix[19] can be used to describe the performance of such a classification model using a test dataset for which the true values are true. Generative modeling refers to the practice of creating artificial instances from a dataset such that they retain similar characteristics to the original set. The model selection metric for the binary classification problem is the AUC-ROC curve. Receiver Operator Characteristic (ROC) for various groups is a probability curve. In terms of the expected likelihood, ROC shows how good the model is for distinguishing the classes given. In this work, it is used for the classification of the normal and pneumonia images. ROC curves are widely used to represent the result of machine learning in binary decision problems.[20] Cross-validation is a tool to evaluate predictive models by splitting the original sample into a training set to train the model and a test set to evaluate it.[21]

3. The Proposed Work

The overall workflow performed in this work is presented in Fig. 1, which is divided into three sub-sections. First, the X-ray image has been used to train the CNN model. Then, DCGAN[3] is used to generate the synthetic X-ray images. Finally, the CNN model is trained using the existing dataset along with the generated synthetic X-ray images.

In the following section, each block of the proposed framework is described in detail.

4. Pneumonia Classification

4.1. *Data*

The data used in this experiment are a chest X-ray images dataset obtained from a Kaggle repository[6] entitled “Chest X-ray Images (Pneumonia)”. The actual size of the dataset is 5863. However, the number of images for pneumonia and normal images were different. So, 1480 pneumonia images were picked from 5863 images to make it equal to normal data. which is also shown in Table 1. For the simplicity of

Int. J. Image Grap. Downloaded from www.worldscientific.com

Int. J. Image Grap. Downloaded from www.worldscientific.com

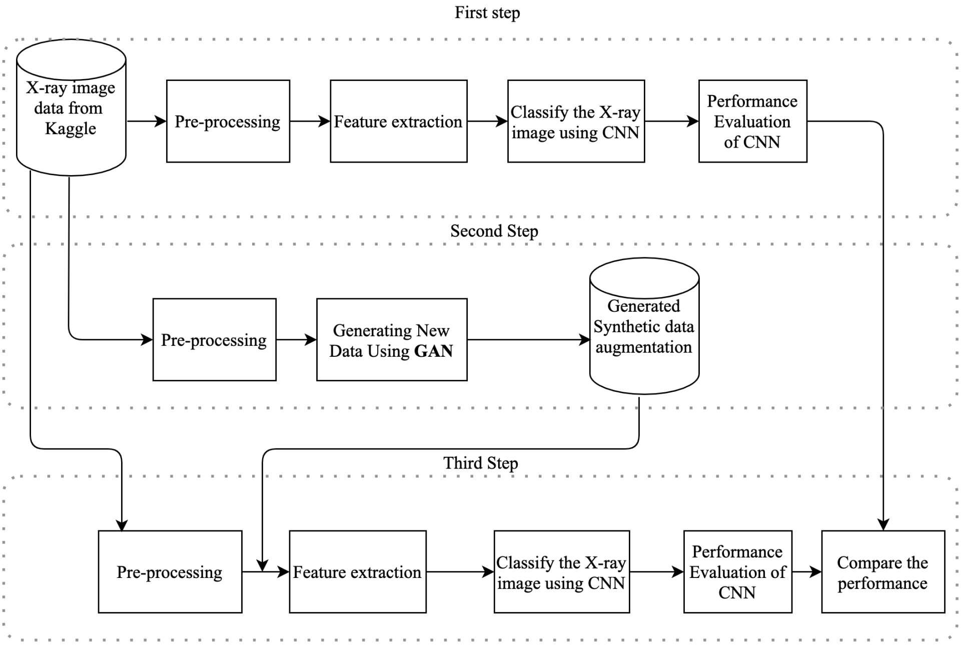
First step
X-ray image data from Kaggle
Pre-processing
Feature extraction
Classify the X-ray image using CNN
Performance Evaluation of CNN
Second Step
Pre-processing
Generating New Data Using GAN
Generated Synthetic data augmentation
Third Step
Pre-processing
Feature extraction
Classify the X-ray image using CNN
Performance Evaluation of CNN
Compare the performance

Fig. 1. Overall workflow diagram of the proposed work.

Table 1. Complete dataset of X-ray images.

Image type	Value
Normal	1480
Pneumonia	1480
Total	2960

training processes, the X-ray images are resized to 128 × 128 pixels, and each pixel value is multiplied by 1/255 to scale them into the range of [0,1].[12] The sample images are presented in Fig. 2.

4.2. *CNN architecture*

The proposed CNN architecture for the pneumonia classification system in this research is shown in Fig. 3. The model consists of five layers of convolutional layers with Max-pooling and ReLU activation function.[15]

5. Generating Synthetic X-ray Images

GAN framework[4] is used to generate visually realistic images. GAN architectures as shown in Fig. 4 are usually explained in terms of game theory or min-max problems,

Int. J. Image Grap. Downloaded from www.worldscientific.com

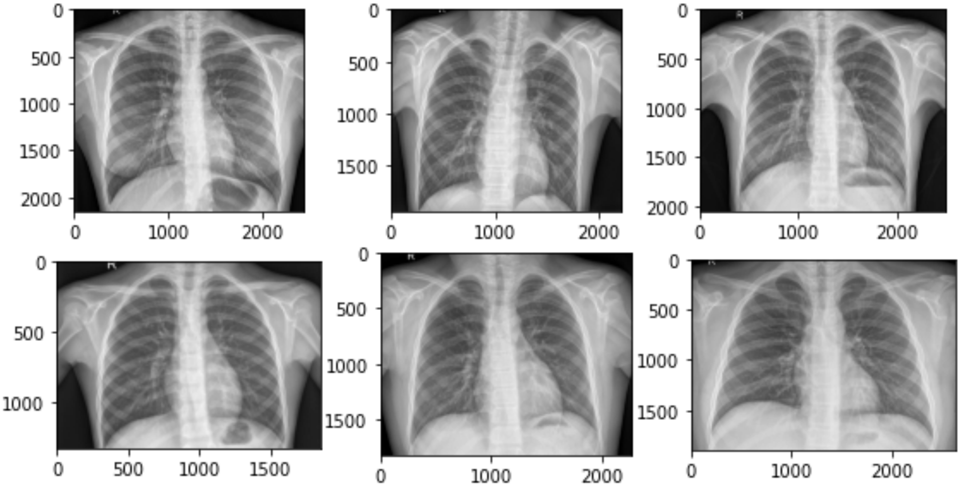

Fig. 2. High-resolution original dataset.

consisting of a generator and a network of discriminators. The generator network produces a sample of similar training images and tries to confuse the discriminator network. The discriminator then distinguishes synthetic images and then feeds its feedback into the generator again, which competes with the generator.[17]

5.1. *DCGAN*

A DCGAN is the advanced version of the GAN,[4] which consists of convolutional and transpose-convolutional networks in the discriminator and generator, respectively. The job of the discriminator D is to maximize the possibility of correctly classifying both training sample and generated samples while the probability is reduced by generator G which is formulated as follows:

$$\begin{aligned}\min\nolimits_G \max\nolimits_D V(D,G) = {} & \min\nolimits_G \max\nolimits_D (E_{xP_{\text{data}}(x)}[\log D(x)] \\ & + E_{zP_z(z)}[\log(1 - D(G(z)))]), \end{aligned} \tag{1}$$

where D is the discriminator, G is the generator, z is the noise vector, $P_z(z)$ is the input noise distribution, $P_{\text{data}}(x)$ is the original data distribution, $G(Z)$ is the generator output for fake data, $D(x)$ is the discriminator output for original data and $D(G(z))$ is the discriminator output for fake data.

In the original DCGAN architecture[3] with Tanh activation function in the generator, ReLU as the activation function in the discriminator and kernels of size 44 were used. In the GAN-Based Synthetic Medical Image Augmentation in Liver Lesion Classification[2] work, the network is configured with a batch size of 64 and Adam optimizer with a learning rate of 2.0×10^{-4}. The image size of 64×64 synthetic liver lesion image has been generated using DCGAN with kernel size 55.

Int. J. Image Grap. Downloaded from www.worldscientific.com

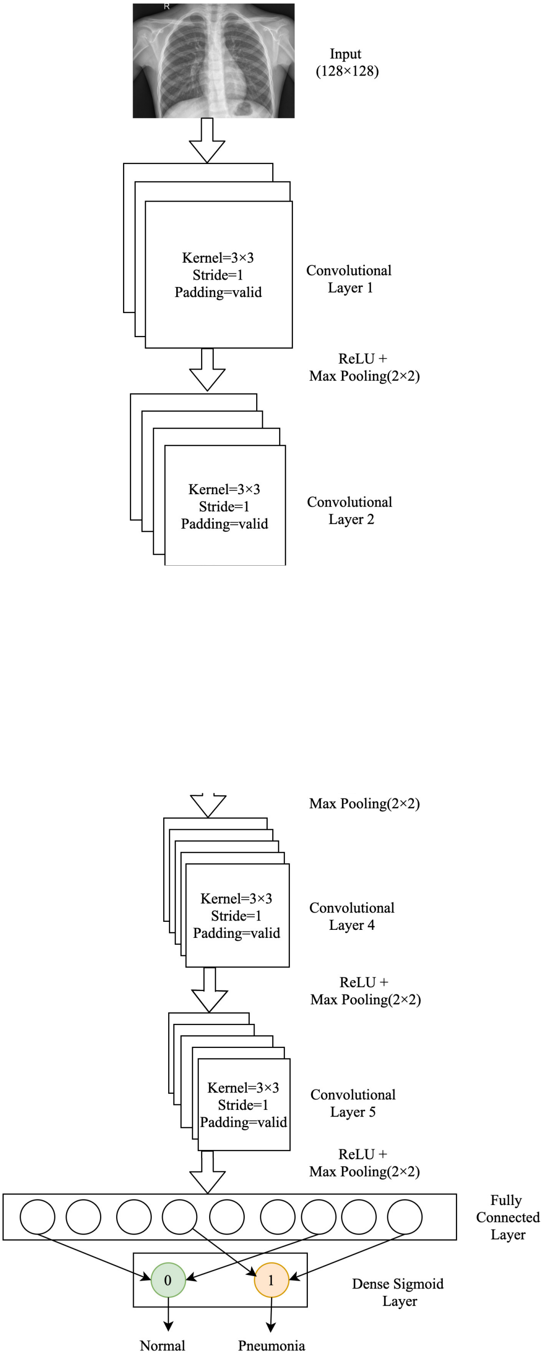

Fig. 3. The architecture of the pneumonia chest X-ray classification CNN.

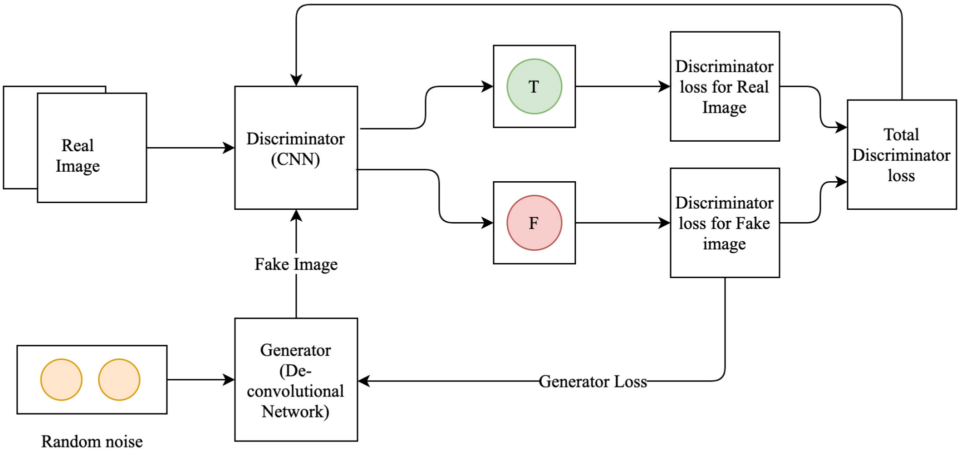
Real Image
Discriminator (CNN)
T
F
Discriminator loss for Real Image
Discriminator loss for Fake image
Total Discriminator loss
Fake Image
Generator (De-convolutional Network)
Generator Loss
Random noise

Fig. 4. Generative Adversarial Network (GAN).

In this work, the X-ray images with the size of 128 × 128 are generated by configuring the hyper-parameters with batch site 128, five convolutional layers, and Leaky ReLU as the activation function in generator.

5.2. *Generator architecture*

The purpose of the generator is to produce images that successfully fool the discriminator. This implies that its objective function aims to allow the model to construct images with the highest possible D value in order to fool the discriminator. The architecture of the generator used to generate the synthetic X-ray images is shown in Fig. 5, where the five convolutional layers are used with the activation function as the Leaky ReLU.[16] The output of the generated image is with a size of 128 × 128.

5.3. *Discriminator architecture*

The discriminator architecture determines whether the image is real or fake. The discriminator is typically a simple CNN in simple architectures.

The discriminator produces a value D that shows how similar the image generated is to the real image. The objective is to increase the probability of the discriminator recognizing real images as real and fake images created by the generator as fake. For this method, the cross-entropy function measures the $p\log(q)$ loss. This function tests a classification model's efficiency. For the classification of the model, the cross-entropy features works especially well because the loss will increase the more the expected likelihood diverges from the mark. The architecture of the discriminator used in this research is shown in Fig. 6, where the five convolution layers with the

Int. J. Image Grap. Downloaded from www.worldscientific.com

Int. J. Image Grap. Downloaded from www.worldscientific.com

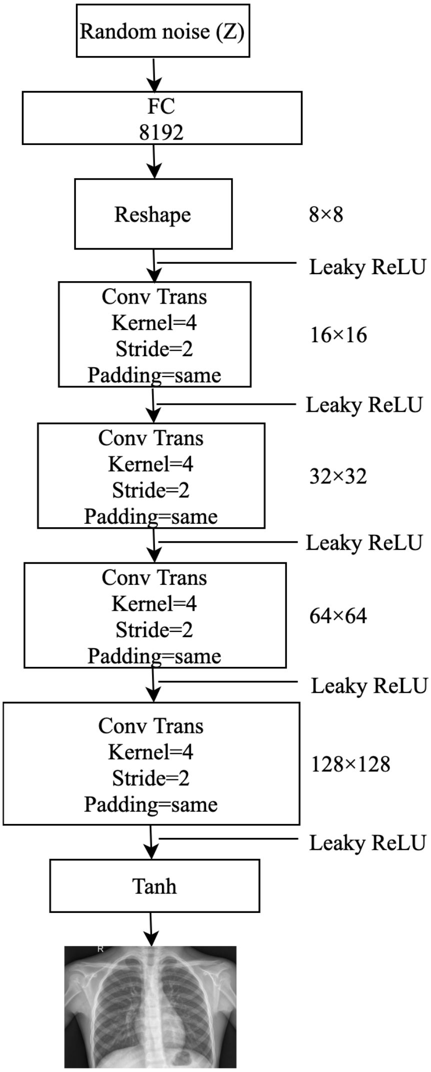

Fig. 5. Generator architecture.

Leaky ReLU as the activation function are used to determine whether the given input image is a real X-ray image or the fake X-ray image generated by the generator.

6. Experiments and Result

6.1. *Training CNN classifier with the existing dataset*

In this section, the CNN model is trained with the existing datasets. The hyper-parameters set to train the CNN are shown in Table 2, where five convolutional

Int. J. Image Grap. Downloaded from www.worldscientific.com

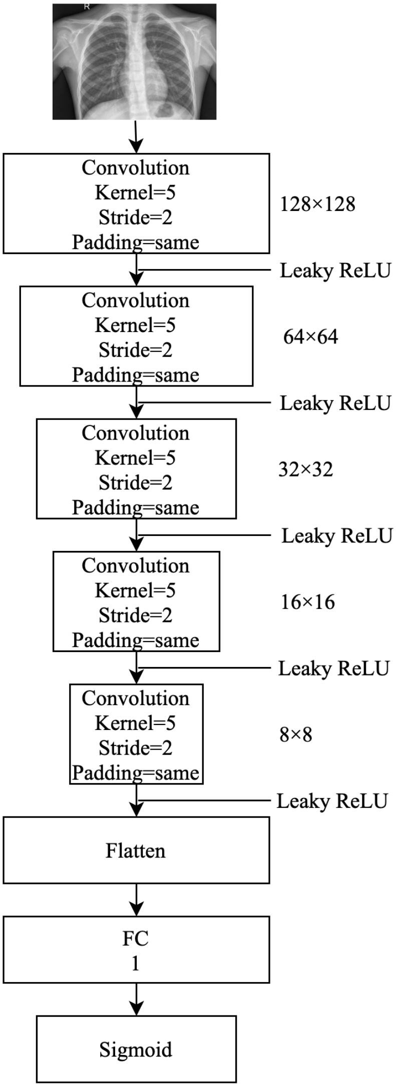

Fig. 6. Discriminator architecture.

layers with the Max pooling 22 and ReLU as the activation function have been used. The model has been trained with 10 epochs.

6.1.1. *Splitting datasets into 60/20/20*

The existing dataset is divided into training, testing and validation sets in the ratio of 60%, 20% and 20%, respectively, which is also shown in Table 3. With this segmentation, training accuracy and validation accuracy are found to be 87.27% and 87.71%, respectively.

Int. J. Image Grap. Downloaded from www.worldscientific.com

Table 2. Training parameters setup for CNN with existing dataset.

Parameter	Value	Parameter	Value
Image size	128 × 128	Convolution layer	5
Max pooling	22	Batch size	64
Step per epochs	20	Epochs	10
Dropout	0.2	Activation function	ReLU/sigmoid

Table 3. Data segmentation for training, testing and validation in the ratio of 60%, 20% and 20%.

Parameter	Value	Parameter	Value
Train normal image	895	Train pneumonia image	895
Val normal image	298	Val pneumonia image	298
Test normal image	298	Test pneumonia image	298

The Confusion matrix for the data segmentation into the ratio of 60%, 20% and 20% is shown in Table 4.

The accuracy and misclassification rate have been found as 86.40% and 0.13%, respectively.

Accuracy: (TP + TN)/Total = 515/596 = 86.40
Misclassification rate: (FP + FN)/Total = 81/596 = 0.13

The training accuracy and validation accuracy of the CNN model and details of loss are shown in Fig. 7.

6.1.2. *Splitting datasets into 70/15/15*

As shown in Table 5, the existing dataset is divided into the training, testing and validation sets in the ratio of 70%, 15% and 15%, respectively. The training accuracy and validation accuracy are found as 88.91% and 90.61%, respectively.

After training the model, the trained CNN model has been tested with a testing dataset with the size each with 223 of normal and pneumonia. The confusion matrix of the same has been shown in Table 6. Then the accuracy and loss were found as follows:

Accuracy: (TP + TN)/total = 392/446 = 0.8789
Misclassification Rate: (FP + FN)/total = 54/446 = 0.121

The training accuracy, validation accuracy and loss for 70/15/15 split are shown in Fig. 8.

Table 4. Confusion matrix for data segmentation into the ratio of 60%, 20% and 20%.

	Predicted:Normal	Predicted:Pneumonia	Total
Actual:Normal	TN = 280	FP = 18	298
Actual:Pneumonia	FN = 63	TP = 235	298
Total	343	253	596

Int. J. Image Grap. Downloaded from www.worldscientific.com

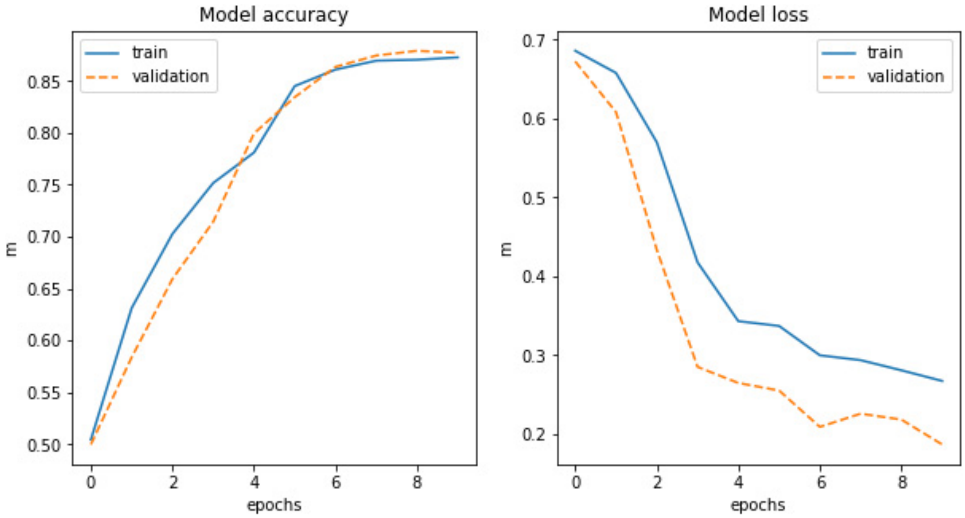

Fig. 7. Training accuracy and loss for data segmentation into the ratio of 60%, 20% and 20%.

6.1.3. *5-fold cross-validation*

From the above analysis, it is found that the best segmentation ratio for this dataset is 70%, 15% and 15%. So, for the 5-fold cross-validation, the existing dataset is segmented into the training, testing and validation sets in the ratio of 70%, 15% and 15%, respectively, which is also shown in Fig. 7.

Training Accuracy Mean = 0.8844
Validation Accuracy Mean = 0.8859
Testing Accuracy Mean = 0.8788
Training Accuracy Standard Deviation = 0.007855
Validation Accuracy Standard Deviation = 0.011198
Testing Accuracy Standard Deviation = 0.007634

6.1.4. *AUC/ROC-based evaluation*

For the existing dataset, the AUC has been found as 0.944. It illustrates that there is a 94% chance that our model successfully differentiates between positive and negative classes. The ROC curve is shown in Fig. 9.

Table 5. Data segmentation for training, testing and validation in the ratio of 70%, 15% and 15%.

Parameter	Value	Parameter	Value
Train normal image	1043	Train pneumonia image	1043
Val normal image	223	Val pneumonia image	223
Test normal image	223	Test pneumonia image	223

Int. J. Image Grap. Downloaded from www.worldscientific.com

Table 6. Confusion matrix for data segmentation into the ratio of 70%, 15% and 15%.

	Predicted:Normal	Predicted:Pneumonia	Total
Actual:Normal	TN = 212	FP = 11	223
Actual:Pneumonia	FN = 43	TP = 180	223
Total	257	189	446

6.2. *Training the GAN to generate the synthetic image*

The hyper-parameter configurations for the generator and discriminator network are shown in Tables 8 and 9. The model has been trained up to 350 epochs. The 1192 synthetic normal images and 1192 pneumonia images of size of 128 × 128 have been generated.

The samples of the generated normal images are shown in Fig. 10 and pneumonia images are shown in Fig. 11.

As shown in Fig. 12, in the early stages of training, both the generator and discriminator are weak in their respective work. After training it for some epochs, the generator becomes better at generating images that can fool the discriminator, and also the discriminator becomes better at classifying the real and fake images. As the discriminator and generator are competitive networks, two components constantly compete with each other and try to become stronger, minimizing the opponents performance.

In the previous research work,[2,3] only the image of the size with 64 × 64 was generated using DCGAN, and four convolution layers were used. In this

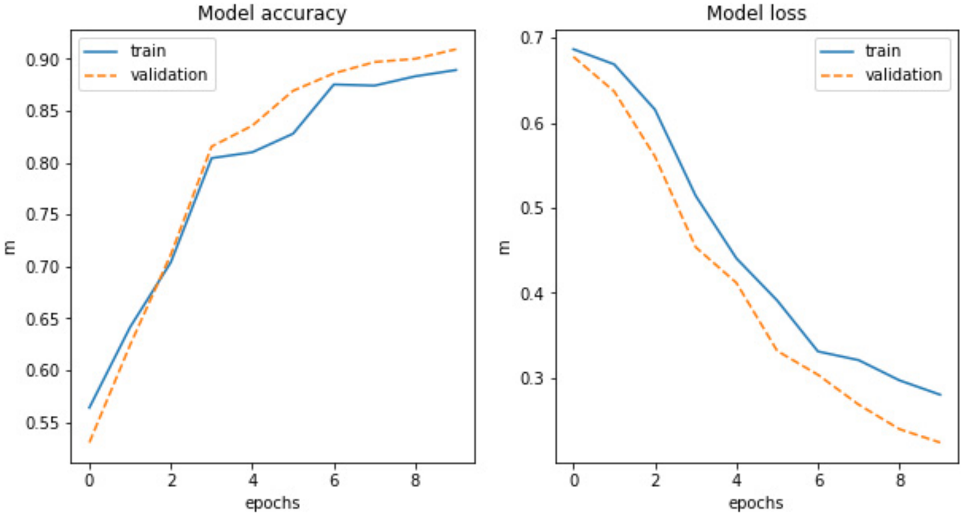

Fig. 8. Training accuracy and loss for data segmentation into the ratio of 70%, 15% and 15%.

Int. J. Image Grap. Downloaded from www.worldscientific.com

Table 7. 5-fold cross-validation for the existing dataset.

Fold	Training accuracy	Validation accuracy	Testing accuracy
1	0.8891	0.8825	0.8891
2	0.8927	0.8801	0.8789
3	0.8892	0.9081	0.8811
4	0.8712	0.8779	0.8654
5	0.8799	0.8809	0.8795

research work, medical images with the size of 128 × 128 have been generated by using DCGAN with activation function as Leaky ReLU and five convolutional layers.

6.3. *Training the CNN model with synthetic images*

By using the DCGAN model, 1192 pneumonia and 1192 normal images are generated. The CNN model has been trained with the addition of an augmented dataset. The parameters setup for the CNN model have been shown in Table 10.

6.3.1. *Train the CNN model with 50% of the generated images*

The CNN model with the same architecture as before has been trained by adding the synthetically generated images. The splitting of train, test and validation dataset is shown in Table 11.

The training accuracy and validation accuracy were found to be 90.27% and 91.09%, respectively. The model accuracy and loss has been shown in the Fig. 13. The calculation for the testing accuracy and loss is performed using a confusion matrix shown in Table 12.

After training the model, the trained CNN model has been tested with a testing dataset with size each with 223 normal and pneumonia images. Then the accuracy

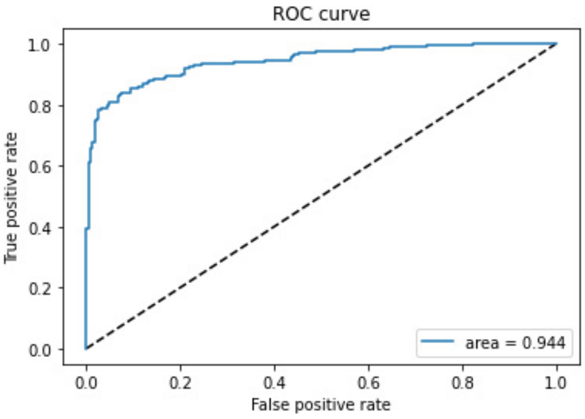

Fig. 9. AUC (Area under the ROC curve) for existing dataset.

Int. J. Image Grap. Downloaded from www.worldscientific.com

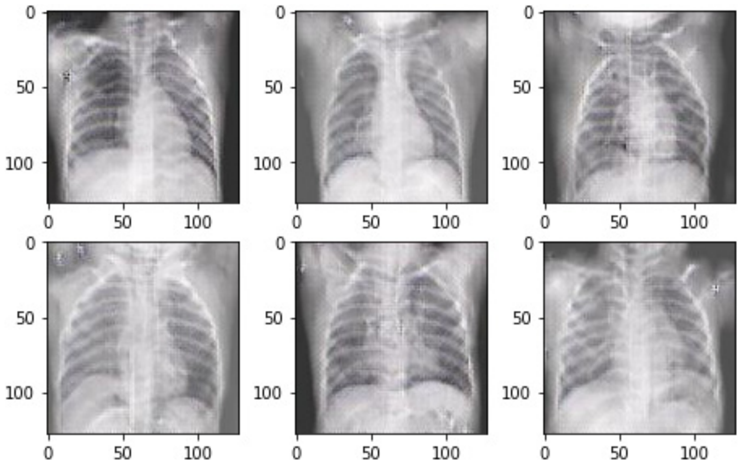

Fig. 10. Generated normal images of size 128 × 128.

and loss are found as follows:

Accuracy: $(\mathrm{TP} + \mathrm{TN})/\mathrm{total} = 401/446 = 0.8991$
Misclassification Rate: $(\mathrm{FP} + \mathrm{FN})/\mathrm{total} = 61/596 = 0.1008$

6.3.2. *Train the CNN model by adding 100% of the generated images*

The CNN model has been trained with the addition of the augmented dataset. The training and validation dataset setup has been shown in Table 13.

The training accuracy and validation accuracy are found to be 91.18% and 91.56%, respectively. The graph of the same has been shown in Fig. 14.

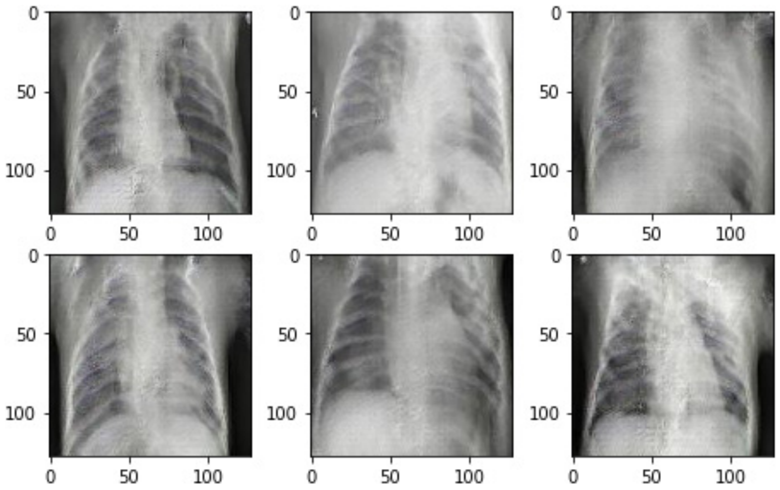

Fig. 11. Generated pneumonia images of size 128 × 128.

Int. J. Image Grap. Downloaded from www.worldscientific.com

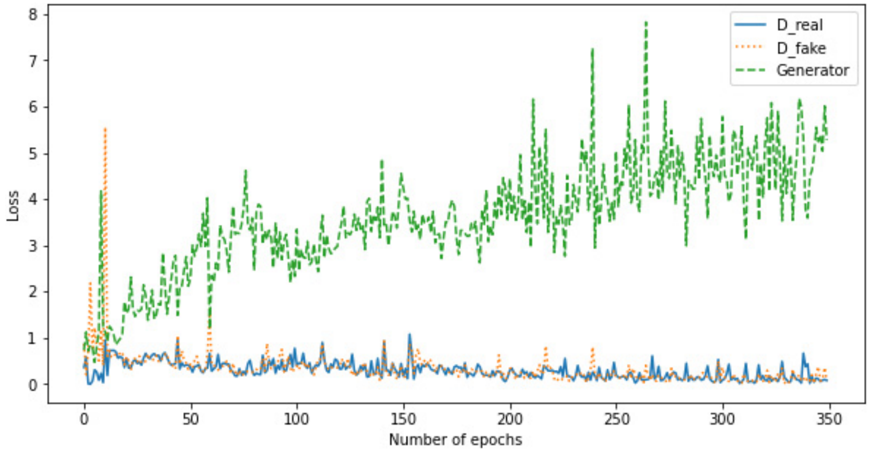

Fig. 12. Generator and discriminator loss during training of DCGAN model.

Table 8. Training parameters setup for generator.

Parameter	Value	Parameter	Value
Normal image	1191	Pneumonia image	1191
Image size	128 × 128	Trans convo layer	5
Kernel size	44	Strides	22
Batch size	128	Latent space	100
Padding	Same	Epochs	350
Learning rate	0.0002	Activation function	Leaky ReLU/tanh

The Confusion matrix for CNN model train with actual dataset and augmented dataset is shown in Table 14. And the Accuracy and misclassification rate are calculated as:
Accuracy: $(\text{TP} + \text{TN})/\text{total} = 407/446 = 0.9125$
Misclassification Rate: $(\text{FP} + \text{FN})/\text{total} = 39/446 = 0.087$

6.3.3. *5-fold cross-validation*

The expanded dataset is segmented the training sets, validation sets and testing sets in the ratio of 70%, 15% and 15%, respectively, for the 5-fold cross-validation, which is also shown in Table 15.

Table 9. Training parameters setup for discriminator.

Parameter	Value	Parameter	Value
Normal image	1191	Pneumonia image	1191
Image size	128 × 128	Convolution layer	5
Kernel size	55	Strides	22
Batch size	128	Input image	128 × 128
Padding	Same	Epochs	350
Learning rate	0.0002	Activation function	Leaky ReLU/sigmoid
Optimizer	Adam	Dropout	0.4

Int. J. Image Grap. Downloaded from www.worldscientific.com [redacted]

Table 10. Training parameters setup for CNN with existing dataset and synthetic images.

Parameter	Value	Parameter	Value
Image size	128 × 128	Convolution layer	5
Max pooling	22	Batch size	64
Step per epochs	20	Epochs	10
Dropout	0.2	Activation function	ReLU/sigmoid

Table 11. Adding the 50% of the generated image in the existing dataset.

Parameter	Value	Parameter	Value
Train normal image	1564	Train pneumonia image	1564
Val normal image	334	Val pneumonia image	334
Test normal image	223	Test pneumonia image	223

Training Accuracy Mean = 0.9158
Validation Accuracy Mean = 0.9154
Testing Accuracy Mean = 0.91082
Training Accuracy Standard Deviation = 0.007757
Validation Accuracy Standard Deviation = 0.004289
Testing Accuracy Standard Deviation = 0.006217

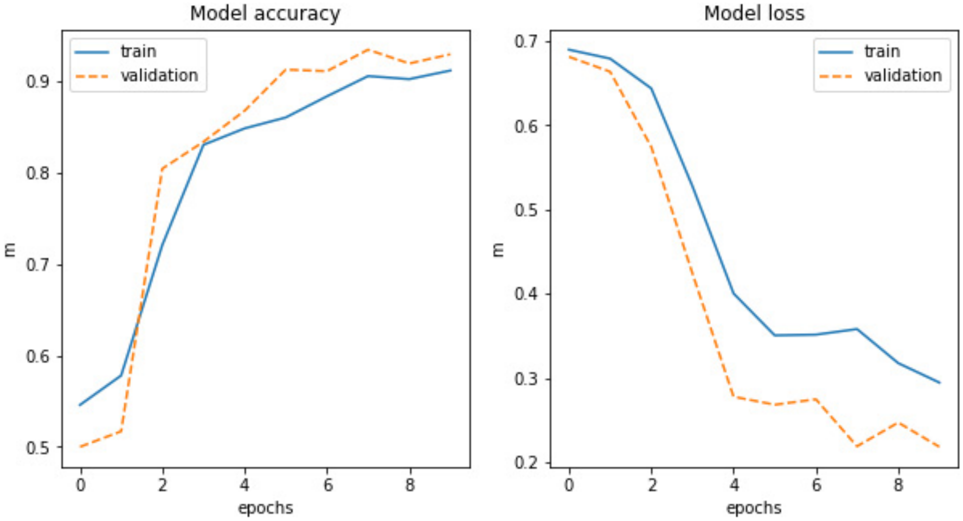

Fig. 13. CNN model accuracy and loss with addition of 50% generated images.

Int. J. Image Grap. Downloaded from www.worldscientific.com

Table 12. Confusion matrix for CNN model train with actual dataset and augmented dataset.

	Predicted:Normal	Predicted:Pneumonia	Total
Actual:Normal	TN = 214	FP = 9	223
Actual:Pneumonia	FN = 36	TP = 187	223
Total	250	196	446

Table 13. Adding the 100% of the generated image in the existing dataset.

Parameter	Value	Parameter	Value
Train normal image	1998	Train pneumonia image	1998
Val normal image	460	Val pneumonia image	460
Test normal image	223	Test pneumonia image	223

Table 14. Confusion matrix for CNN model train with actual dataset and augmented dataset.

	Predicted:Normal	Predicted:Pneumonia	Total
Actual:Normal	TN = 216	FP = 7	223
Actual:Pneumonia	FN = 32	TP = 191	223
Total	248	198	446

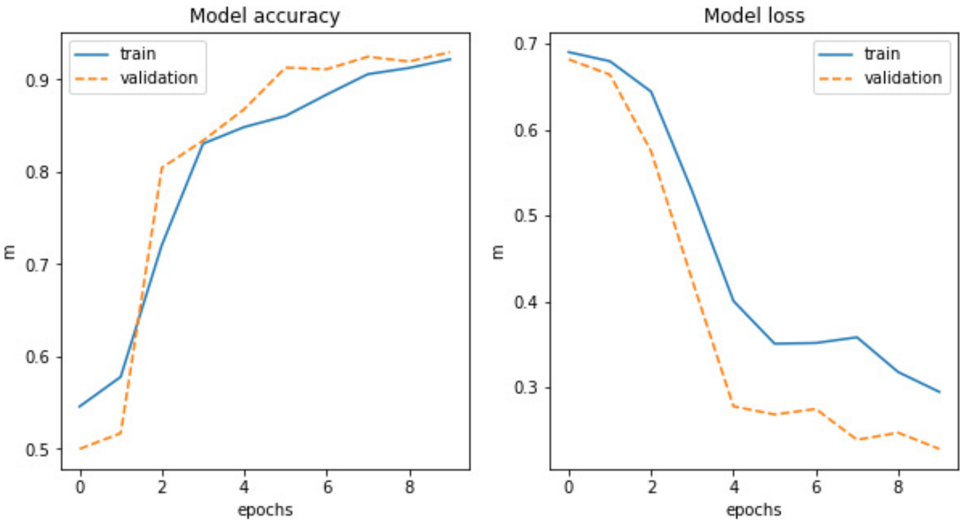

Fig. 14. CNN model accuracy and loss with addition of generated images.

Table 15. 5-fold cross-validation and testing.

Fold	Training accuracy	Validation accuracy	Testing accuracy
1	0.9118	0.9156	0.9146
2	0.9209	0.9101	0.9108
3	0.9089	0.9115	0.9040
4	0.9091	0.9186	0.9049
5	0.9287	0.9216	0.9207

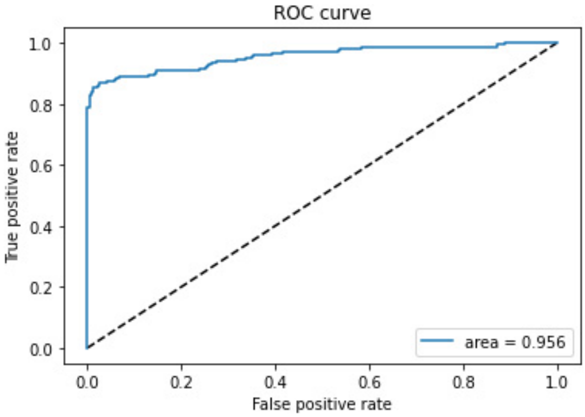

Fig. 15. Area under the ROC curve after adding synthetic images.

6.3.4. *AUC/ROC base evaluation*

After adding the synthetic images to train the CNN model, the ROC curve is seen to be improving, and the AUC has been found to be 0.956. It shows that there is a 95% chance that the model will successfully differentiate between positive class and negative class. The same concept is illustrated in Fig. 15.

7. Conclusion and Future Work

This research presents an approach of using synthetic images to enhance the performance of the CNN model with the limited dataset. The DCGAN model is implemented to generate the synthetic medical images for data augmentation. The proposed augmentation technique on pneumonia and normal image classification task has achieved an improvement of 3.2% using over the classic augmentation techniques. Training and testing a CNN model on a dataset containing both the generated and unique images give greater accuracy than merely implementing the CNN model on a dataset with the original images only.

This study still has room for progress. One possible extension could be an increase in the resolution of the generated image so that the minute features of the images are preserved. DCGANs seem to be extremely sensitive to hyper-parameter choice, and a

Int. J. Image Grap. Downloaded from www.worldscientific.com

lot of problems occurred during training; the quality of the synthetic image we obtained could possibly be enhanced by carefully tuning the hyper-parameters in the network.

Acknowledgments

We would like to express our great appreciation to Prof. Dr. Shashidhar Ram Joshi, Dean, Institute of Engineering, for his valuable and constructive guidance throughout the duration. His willingness to give his continuous support is greatly appreciated.

References

1. G. Verma and S. Prakash, "Pneumonia classification using deep learning in healthcare," *Int. J. Innov. Technol. Exploring Eng.* **9**(4), 1716 (2020).
2. M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger and H. Greenspan, "Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Int. J. Innov. Technol. Exploring Eng.* 1–10 (2018).
3. A. R. L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks" (2016), arXiv:1511.06434v2.
4. I. J. Goodfellow, "Generative adversarial nets" (2014), arXiv:1406.26612.
5. J. Gua and Z. Wangb, "Recent advances in convolutional neural networks" (2017), arXiv:1512.07108v.
6. P. Mooney, "Chest X-ray images (pneumonia)" (2018), Available at https://www.kaggle.coms.
7. H. A. Shiddieqy, T. Adiono and F. I. Hariadi, "Implementation of deep-learning based image classification on single board computer," *Int. Symp. Electronics and Smart Devicess* (2017), pp. 133–137.
8. A. Aurisano, A. Radovic, D. Rocco, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa and P. Vahle, "A convolutional neural network neutrino event classifier" (2016).
9. K. E. Asnaoui, Y. Chawki and A. Idri, "Automated methods for detection and classification pneumonia based on x-ray images using deep learning" (2020).
10. F. H. K. dos Santos Tanaka and C. Aranha, "Data augmentation using gans" (2019).
11. 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 Trans. Med. Imaging* **35**(5), 1153–1159 (2016).
12. S. Kotsiantis, D. Kanellopoulos and P. E. Pintelas, "Data preprocessing for supervised leaning," *Int. J. Comput. Sci.* **1**(1), 111–113 (2006).
13. S. Kulkarni and S. Harnoorkar and P. E. Pintelas, "Comparative analysis of CNN architectures," *Int. Res. J. Eng. Technol.* (2020).
14. M. Mishra and M. Srivastava, "A view of artificial neural network," *Int. Conf. Advances in Engineering and Technology Research* (2014).
15. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *International Conference on Engineering and Technology* (2017).
16. B. Xu, N. Wang, T. Chen and M. Li, "Empirical evaluation of rectified activations in convolution network" (2015), arXiv:1505.00853v2 [cs.LG].
17. C. Han, H. Hayashi, L. Rundo, R. Araki, W. S. S. Muramatsu, Y. Furukawa, G. Mauri and H. Nakayama, "Gan-based synthetic brain mr image generation," *IEEE 15th Int. Symp. Biomedical Imaging* (2018).

Int. J. Image Grap. Downloaded from www.worldscientific.com

Int. J. Image Grap. Downloaded from www.worldscientific.com

18. I. Sergey and S. Christian, “Batch normalization: Accelerating deep network training by reducing internal covariate shift” (2015).
19. B. P. Salmon, W. Kleynhans and C. P. S. J. C. Olivier, “Proper comparison among methods using a confusion matrix,” *IEEE Int. Symp. Geoscience and Remote Sensing* (2015).
20. J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proc. 23rd Int. Conf. Machine Learning*, June 2006.
21. J. D. Rodrguez, A. Perez and J. A. Lozano, “Sensitivity analysis of k-fold cross validation in prediction error estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.* (2010).
22. D. Mahapatra, B. Bozorgtabar, S. Hewavitharanage and R. Garnavi, “Image super resolution using generative adversarial networks and local saliency maps for retinal image analysis,” *Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, September 2017.
23. T. Schleg, P. Seebock, S. M. Waldstein, U. Schmidt-Erfurth and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery” (2017), arXiv:1703.05921v1 [cs.CV].
24. D. Nie, R. Trullo, C. Petitjean, S. Ruan and D. Shen, “Medical image synthesis with context-aware generative adversarial networks,” University of North Carolina at Chapel Hill, USA (2016).
25. O. A. Ramwala, S. A. Dhakecha, C. N. Paunwala and M. C. Paunwala, “Reminiscent net: Conditional gan-based old image de-creasing,” *Int. J. Image Graphics* (2021).
26. C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data* (2019).
27. R. Khan and R. Debnath, “Morphology preserving segmentation method for occluded cell nuclei from medical microscopy image,” *Int. J. Image Graphics* (2021).
28. K. K. Jayaraman, K. Srilakshmi and S. Jayaraman, “Modified flower pollination-based segmentation of medical images,” *Int. J. Image Graphics* (2021).

Ramesh Adhikari received his Bachelor’s and Master’s degrees, both in computer engineering, from the Tribhuvan University and Pokhara University, Nepal, in 2017 and 2021, respectively. He is currently in the Technical Innovation Department at Sanima Bank, Nepal. His research interests include computer vision, pattern recognition and machine learning.

Suresh Pokharel received his Bachelor in Computer Engineering and M.Sc. in Computer Systems and Knowledge Engineering degrees from the Tribhuvan University, Nepal. He is currently a PhD student in the College of Computing at the Michigan Technological University. His research interests include bioinformatics, computer vision and deep Llearning algorithms.