Manila Journal of Science ***14*** *(2021), pp. 35–54*

Detection of Pneumonia in Chest X-Ray Images Using Deep Transfer Learning and Data Augmentation With Auxiliary Classifier Generative Adversarial Network

Christi Florence Cala-or,[1,*] Ara Abigail Ambita,[1] Almie Carajay,[1,2] and Joanah Faith Sanz[1]

[1]Division of Physical Sciences and Mathematics, University of the Philippines Visayas, Miagao, Iloilo, Philippines
[2]Philippine Genome Center Visayas, Miagao, Iloilo, Philippines

*Email: cccalaor@up.edu.ph

ABSTRACT

Deep learning applications in medical research are often constrained by the lack of data availability due to the significant labor and cost required to collect data. Such issues cause the convolutional neural networks (CNNs) to suffer with overfitting and a drastic loss in accuracy. To overcome this problem, generative adversarial networks (GANs) have been adopted in medical imaging as a data augmentation technique because of their capability to generate realistic samples that help add variability in the training set. Therefore, this paper proposes a data augmentation based on GAN to overcome the issue of limited data availability in conjunction with pretrained CNN models on detecting pneumonia from chest x-ray images. We use auxiliary classifier GAN (ACGAN), which extends traditional GAN by making the generation of images conditional on a side information such as labels. The proposed method has further improved the performance of the CNN models most especially the ResNet variants that improved by more than 10%. ResNet-18, the smallest ResNet variant, showed the highest improvement with 13.36% in accuracy and 16.13% in F_1-score and also outperformed the other CNN models used in the experiment. The addition of ACGAN-generated images has proven to be effective in adding variability to the training set.

Keywords: machine learning, GAN, ACGAN, deep learning, CNN, transfer learning

INTRODUCTION

According to the World Health Organization (WHO), pneumonia is one of the leading causes of death among children under 5 years old and elderly worldwide, killing around 808,694 children in 2017 ("Pneumonia," n.d.). It is a form of acute respiratory infection caused by a virus, bacteria, fungi, or other pathogens that affects the small air sacs (alveoli) in the lungs ("Learn About Pneumonia," n.d.). Pneumonia results in inflammation in the lungs that can be life-threatening if not diagnosed early.

Different imaging modalities such as chest x-ray, computed tomography (CT), and magnetic resonance imaging (MRI) are used to diagnose pneumonia and other lung diseases. Chest x-ray is the most common method to detect pneumonia since it is an economical and easy-to-use medical imaging and diagnostic technique.

However, detecting pneumonia from chest x-rays is still largely dependent on the diagnostic level of the radiologist, and the reliability of the results is challenging even for highly experienced radiologists as these images have similar opacities for other various lung abnormalities such as lung cancer and excess fluid (Li et al., 2020). Moreover, it is more challenging especially in the African and South Asian countries where trained personnel are lacking and medical resources are limited (Liang & Zheng, 2020).

Advancements in deep learning have improved the performance of healthcare professionals in various imaging modalities including pneumonia detection. It has been used in replacement for conventional computer-aided detection (CADs) in classifying chest x-ray images (Li et al., 2020). The use of deep convolutional neural networks (CNNs) to diagnose pneumonia using chest x-ray has been a powerful tool for computer vision tasks that include large data sets because of its powerful computational capabilities (Jain et al., 2020).

In order to achieve superior performance with deep learning, a large amount of training data is required. The bigger the size of the data set, the more accurate the model becomes (Jain et al., 2020). However, due to the limited availability of training data, CNNs suffer with overfitting and a drastic loss in accuracy. This persisting challenge of limited data availability and consequently a constrained power of deep learning is prevalent in medical data analysis or imaging wherein the collection of medical data often requires significant labor, complex and expensive collaboration of researchers and radiologists, and funding (Albert, 2020; Shaikhina & Khovanova, 2017). Although there are free and accessible public medical data sets, they still suffer with limited size and are incapable of generalizing in other data sets (Frid-Adar et al., 2018). This is because deep learning models are task specific and can no longer recognize features that are outside the training domain.

To overcome these limitations, data augmentation techniques such as translation, rotation, flip, or scale have been used to stabilize the training process and to reduce overfitting. Data augmentation provides more possible data points that could possibly minimize the distance between the training and validation sets (Albert, 2020).

Another data augmentation technique is by using generative adversarial networks (GANs) to generate synthetic samples that

can be added to the original data set. In contrast to the previously mentioned classic data augmentation techniques, which provide only relatively minimal modifications, GANs can generate realistic images that could potentially add variability to the training set (Frid-Adar et al., 2018). This capability of GANs has attracted researchers in medical imaging and has been adopted in many applications such as low-dose CT denoising (Kim et al., 2020; Wolterink et al., 2017), skin lesion synthesis (Frid-Adar et al., 2018), organ segmentation (Dong et al., 2019), and cross-modality transfer such as MR to CT (Qian et al., 2020).

Despite the promising approach of GAN for image synthesis, GANs struggle to generate high-resolution samples—particularly from data sets with high variability. One variant of GAN, the conditional GANs, aims to overcome this challenge by adding a side information or adding class labels to improve the generated image's quality (Mirza & Osindero, 2014). Also, conditional GANs (cGAN) have been widely adopted in medical imaging, particularly on MRI and CT scan images. cGAN is the base concept of the multiconditional generation of realistic and diverse nodules placed naturally on lung CT at desired position/size/attenuation, which even expert physicians cannot distinguish from the real ones (Han et al., 2019). Cross-domain synthesis, for example, multicontrast abdomen MRI synthesis from corresponding CT images based on cGAN, was also showed to increase the diversity of diagnostic information as well as improve registration and segmentation tasks (Yang et al., 2019). Similar results have also been reported where the performance of cGAN with different generator architectures and MRI scanners for magnetic resonance to synthetic computer tomography (MR-sCT) conversion has been investigated (Fetty et al., 2020). More importantly, the use of cGAN architecture was shown to be useful despite the small sample size used for the generator-discriminator cross-training.

The quality of cGAN for image synthesis can actually be further improved by adding an auxiliary classifier. Auxiliary classifier GAN (ACGAN) further extends cGAN by making the discriminator not only classify whether the image is real but also classify the class/label of the image (Odena et al., 2017). Similarly, medical imaging problems such as CT from MRI abdominal image synthesis have also been explored with ACGAN (Qian et al., 2020). This method was shown to be capable and robust in estimating superior CT scans with quite a limited sample.

In addition, ACGAN was also adopted for artificially extending the data set by generating synthetic images particularly for improving CT and x-ray image classification tasks. For instance, ACGAN was employed to generate synthetic pediatric CT scans since the data are hard to obtain due to risks of exposing children to radiation (Kan et al., 2020). Images were conditionally synthesized with a vector denoting the desired age classes.

Another study investigated ACGAN in conjunction with CNN for detection of Covid-19 in chest x-ray images. Since the pandemic is recent and the data set is relatively small, the generation of synthetic samples was employed to ramp up the limited number of chest x-ray images available for study. ACGAN-generated samples have added

variability to the original data set, which substantially improved their classification accuracy by 10% (Waheed et al., 2020).

The lack of data in medical imaging, particularly for the detection of pneumonia in chest x-ray images, led us to explore other ways to expand our data set. Motivated by the mentioned studies, we mainly focus on employing GAN for data augmentation to generate synthetic chest x-ray images as additional samples. As these synthetic images introduce variability in the data set, the predictive capacity of the deep learning methods we will utilize for this classification will also improve.

Transfer learning will be utilized to reduce the time taken to develop and train a model. Essentially, it is a process where we reuse the weights of already existing models (Theckedath & Sedamkar, 2020). We combine synthetic chest x-ray images generated using ACGAN with these transfer learning models. This research then has the following contributions:

1. Propose an auxiliary classifier adversarial network (ACGAN) for data augmentation to generate additional synthetic chest x-ray samples to overcome the limited data availability.
2. Utilize pretrained CNN models such as VGG-16, DenseNet-121, ResNet-18, ResNet-50, ResNet-101, and ResNet-152 for the detection of pneumonia in pediatric chest x-ray images.
3. Improve the classification performance of these pretrained CNN models for the pneumonia detection by combining the generated synthetic images with the original training set.

The methods that we used aim to find a solution and improve the existing models that have been used to solve this specific problem by introducing variability in the data set, rather than compete for higher accuracy of those models.

The rest of the paper is as follows: the details of the methods such as CNNs and ACGAN, data set, training and implementation, metrics, and tools are described in the Materials and Methods section of this paper. The experimental results including the performance of CNN models with and without ACGAN using different hyperparameters are discussed in the Results and Discussion section. And finally, we discuss the insights we have gathered from the experiments as well as the set of limitations that we aim to address in the future.

MATERIALS AND METHODS

In this section, we present the details of the proposed model, preprocessing applied, data set, and the metrics used to evaluate the classification performance. We propose the use of ACGAN to generate synthetic samples to alleviate the issue of limited data availability. For the detection of pneumonia, pretrained CNN models such as VGG-16, DenseNet-121, ResNet-18, ResNet-50, ResNet-101, and ResNet-152 have been used.

The realistic synthetic samples are combined with the original training set in training the model. We then evaluate the impact of ACGAN by comparing the performance of CNN models with and without ACGAN. The flow of the process for the proposed method is shown in Figure 1.

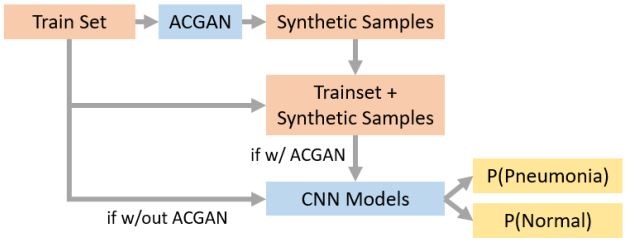

Figure 1. Flowchart of the proposed method.

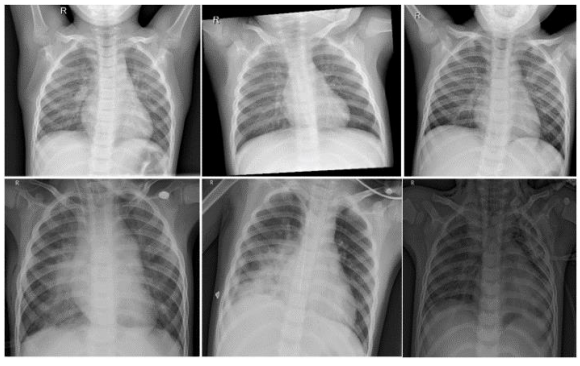

Figure 2. Images with varying intensities and orientations (top 3 images are under normal conditions and the lower 3 are pneumonia cases).

Data Set

The data used for this study consist of 5,856 chest x-ray images (anterior-posterior) of pediatric patients that were collected and labeled by Kermany et al. (2018), of which 4,273 were labeled as positive cases (infected with pneumonia) and 1,583 as negative cases (normal). The data set originated from a total of 5,863 pediatric patients of Guangzhou Women and Children's Medical Center in Guangzhou, in which the chest x-ray imaging was performed as part of the patient's routine clinical care. After careful assessment of experts in the field for chest radiography, each image was screened for quality control; thus, x-ray images with low quality or unreadable scans were removed from the original data set.

The complexity of the classification can be attributed to the varying sizes, orientation, and gray pixel intensities of the x-ray images as shown in Figure 2. It can be observed that in the images of pneumonia cases, the alveoli become filled with secretion (inflammatory fluid) that appears as a white spot in the chest radiograph. The whitish area corresponds to the lung opacity, which characterizes a pulmonary consolidation (Saraiva et al., 2019).

The train-test-split distribution of images, shown in Table 1, is suggested by the author of the data set. Chest x-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients, and it has been ensured that x-ray images of the same patient belong to only one category (training, testing, validation) to prevent data leaking.

Table 1. Data Set Distribution.

	Normal	Pneumonia
Train	1,341	3,875
Test	234	390
Validation	8	8

Generative Adversarial Networks (GANs)

A GAN is a deep CNN introduced by Goodfellow et al. (2020) that can generate images through adversarial processes where two models, the discriminator and the generator, are being trained simultaneously. The discriminator distinguishes real samples

from generated samples, while the generator tries to generate fake samples as real as possible, which makes the discriminator believe that the fake sample is a real one (Mao et al., 2017).

The two models are trained simultaneously with opposite goals. *G* aims to fool the *G*, so it is trained to maximize the final classification error between the real and the generated data. Meanwhile, *D* is trained to minimize the classification error. So, at each iteration of the training process, the weights of *D* are updated to get better at discriminating between real and fake samples in the next round. We use the discriminator loss, which penalizes the discriminator for misclassifying real as fake or generated as real. And more importantly, the weights of *G* are updated based on how well, or not, the generated samples fooled the discriminator. We use the generator loss, which penalizes the generator for failing to fool the discriminator and generating a sample that the discriminator classifies as fake (Goodfellow et al., 2020).

In the perspective of a game, *D* and *G* play a two-player minimax game and try to compete with each other. Equilibrium is reached when the generator produces samples that follow the data distribution and the discriminator predicts whether it is real or fake with equal probability.

Because of these features of GAN, it has been used by many studies since it can make image data sets bigger and can generate impressive results for unsupervised learning tasks.

Auxiliary Classifier GAN (ACGAN)

The basic architecture of GAN consists of two networks trained jointly: a generator *G* and a discriminator *D*. *G* takes a random noise vector *z* from a latent space as input and generates an image $X_{fake} = G(z)$. The latent space is drawn from a gaussian distribution. Through training, the generator learns to map points into the latent space with specific output images, forming a compressed representation of the data distribution (Goodfellow et al., 2020).

Meanwhile, the discriminator *D* receives either a real (training image) or fake (generated image) and outputs a probability distribution $P(S|X)$ over possible image sources where *X* may be X_{real} (real sample) or X_{fake} (generated sample). For instance, $P(S = \text{real} \mid X_{real})$ refers to the probability that the provided image is real given that it is real (from the training domain). The discriminator is trained to maximize the log-likelihood it assigns to the correct source, formally expressed in Equation 1.

$$L = E[\log P(S = \text{real} \mid X_{real})] + E[\log P(S = \text{fake} \mid X_{fake})] \quad (1)$$

The basic GAN framework can be augmented using side information (Odena et al., 2017). A cGAN extends the basic GAN by generating images that are *conditional* on the class label. Essentially, the basic GAN is changed such that the generator is provided with a class label $c \sim p_c$ as an input in addition to the noise *z* or the random point in the latent space. *G* uses both to generate images $X_{fake} = G(c, z)$. The role of the discriminator remains unchanged, that is, to predict whether the image is fake or real.

However, it now also receives the class label as an input.

The auxiliary classifier GAN, or ACGAN, which was introduced by Odena et al. (2017) from the Google Brain, further extends GAN by building upon the idea of cGAN. Similar to cGAN, the generator takes both a noise z from a latent space and a class label. The main modification is the additional role given to the discriminator, which is to output two probabilities. When initially, in basic GAN, the discriminator only outputs $P(S|X)$, ACGAN also outputs $P(C|X)$.

(1) $P(S|X)$—probability distribution over sources (similar to basic GAN), and

(2) $P(C|X)$—probability distribution over class labels.

With this, the objective function will have two parts: the log-likelihood of the provided image, L_S, and the log-likelihood of the correct class, L_C.

$$L_S = E[\log P(S = \text{real} \mid X_{real})] + E[\log P(S = \text{fake} \mid X_{fake})] \quad (2)$$

$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})] \quad (3)$$

During the training process, the discriminator will try to maximize $L_S + L_C$, essentially the probability of classifying real and fake images (L_S) and the probability of correctly identifying the class (L_C). On the other hand, the generator seeks to minimize the ability of the discriminator to discriminate between real and fake images whilst also maximizing the ability of the discriminator predicting the class label of real and fake images (e.g., $L_C - L_S$).

Structurally, the model is still relatively similar to existing models of GANs. However, the slight modification to the GAN training has demonstrated more stable training and consequently generated better images (Odena, 2017).

Figure 3 shows the diagram of how ACGAN is used for this study. The class/label c (pneumonia or normal) and the noise vector z in the latent space serve as inputs to the generator, producing a fake CXR $X_{fake} = G(c, z)$. These images together with the real images are then fed to the discriminator, tasked to predict the $P(C|X)$ or the correct class label of the x-ray image (pneumonia or normal) as well as the $P(S|X)$, the correct source (generated or real from the training domain).

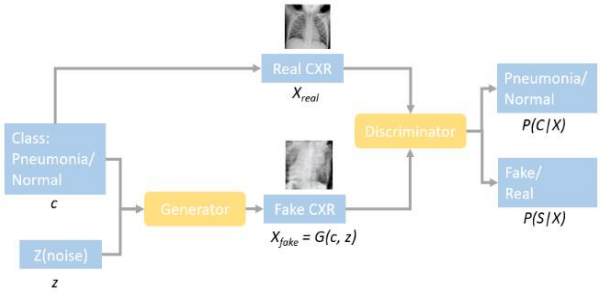

Figure 3. ACGAN model for pneumonia detection.

Results for the $P(S|X)$ and $P(C|X)$ will be dependent on the networks for the generator and discriminator. This can be defined as per the ACGAN architecture (Radford et al., 2016), characterized by using Gaussian weight initialization, batch normalization that simply standardizes the inputs to a layer for each mini-batch for a more stable training, ReLU activation for all layers in the generator except the output layer that uses tanh and in the discriminator

using sigmoid or softmax, and 2 × 2 stride for downsampling.

The ACGAN model has two output layers, to produce $P(S|X)$ and $P(C|X)$ values. The first is a single node with the sigmoid activation to predict whether the image is fake or real, $P(S|X)$, whereas the second has two nodes, one for each class (pneumonia, normal), using the softmax activation function to predict the class label of the given image, $P(C|X)$.

The generator structure of the ACGAN-based image classification model is shown in Table 2.

Table 2. ACGAN Generator and Discriminator Model Configuration

Operation	Kernel	Strides	Feature Maps	Batch Normalization (BN)	Nonlinearity
Gx(z)—130 × 1 × 1					
Transposed convolution	4 × 4	1 × 1	512	Yes	ReLU
Transposed convolution	4 × 4	2 × 2	256	Yes	ReLU
Transposed convolution	4 × 4	2 × 2	128	Yes	ReLU
Transposed convolution	4 × 4	2 × 2	64	Yes	ReLU
Transposed convolution	4 × 4	2 × 2	3	No	Tanh
D(x)—64 × 64 × 3					
Convolution	4 × 4	2 × 2	64	No	LeakyReLU
Convolution	4 × 4	2 × 2	128	Yes	LeakyReLU
Convolution	4 × 4	2 × 2	256	Yes	LeakyReLu
Convolution	4 × 4	2 × 2	512	Yes	LeakyReLu
Convolution	4 × 4	1 × 1	64	No	
Linear	N/A	N/A	2	No	Sigmoid
Linear	N/A	N/A	2	No	Sigmoid

The generator is composed of five transposed convolution layers. The structure of the first transposed convolution layer is (kernel_size is 4, stride is 1), while the rest are (kernel_size is 4, stride is 2). Each layer passes through a batch normalization layer with ReLU as its activation function.

Contrary to the structure of the generator, the discriminator is composed of five convolutional layers, but the kernel and strides are similar to those of the generator. Its outputs are the posterior probability estimation of the sample label, $P(C|X)$, in addition to the probability whether the image is fake or real, $P(S|X)$.

ACGAN Training

For the ACGAN training, we only used the training data of the original data set in order to prevent leaking into the test set and therefore get faulty results. This is composed of 1,341 normal and 3,875 pneumonia training images (see Table 1 for details).

The ACGAN model is trained to synthesize pediatric chest x-rays for both the normal and the pneumonia classes. The image processing involved resizing the image to $64 \times 64 \times 3$ and normalizing the images with (0.5, 0.5, 0.5), (0.5, 0.5, 0.5) values. It is a process by which we change the range of pixels between 0 to 1 in order to help the model converge. Data augmentation techniques such as random horizontal flipping and center crop were employed.

For the training proper, we use the Adam optimizer since it works on sparse gradients, requires little memory space, and is computationally efficient (Waheed et al., 2020). The Adam optimizer is used along with other hyperparameters shown in Table 3. Two loss functions, one for each output layer of the discriminator, were used to optimize the GAN. The first layer uses a binary cross-entropy loss (BCELoss) and the second uses a negative log-likelihood loss (NLLLoss) function.

Table 3. Hyperparameters of ACGAN

Parameter	Value
Max epochs	600
Learning rate	0.002
Batch size	32
Beta	(0.5, 0.999)
Image size	64

At every 20th epoch, we generate a visualization of the generated samples. The issue of having an objective and quantitative approach to evaluate the synthetic images generated by GAN persists in all synthetic medical image generation studies. As a limitation, we simply use human judgment to determine which epoch has generated the clearest images, that is, images with less noisy artifacts and that have closely followed the structure of the chest x-ray. Also, we do not intend to produce accurate or medically correct images that will be used by radiologists of physicians, rather just introduce variability in the images used for training.

In order to address this limitation, the main metric we use to evaluate whether the ACGAN is effective is by observing the accuracy, precision, recall, and F_1-score of the CNNs we have utilized, which will be discussed in the proceeding sections.

Convolutional Neural Network (CNN)

A CNN is a class of deep learning model designed to automatically and adaptively learn the invariant hierarchical features of an input from first learning the low-level features where these features are combined later to learn more complex patterns (Jmour et al., 2018). A CNN model can be achieved by training labeled data and fine-tuning parameters. However, according to Waheed et al. (2020), if CNN is being used in small data sets, there is a high probability of overfitting because of the large number of parameters; therefore, the size of labeled data is proportional to the efficiency of generalization.

In this study, there are three (3) CNN models used, namely, VGG-16, ResNet variants, and DenseNet-121. The VGG-16 network is a CNN model that consists of sixteen (16) convolutional layers with a small receptive field of 3 × 3. Its max pooling layer has a size of 2 × 2 and a total of five layers, where there are three connected layers after the max pooling layer. With these, VGG-16 performs well with image processing (Wani et al., 2020).

Residual Networks (ResNet) variants introduce the concept of residual learning. It predicts the delta that is required to reach the final prediction from one layer to the next (He et al., 2015). ResNet uses the identity mapping that allows the model to bypass a CNN weight layer in the event that the current layer is not vital, which helps in avoiding problems of overfitting to the training set. In this study, we used 18-, 50-, 101-, and 152-layer ResNet (ResNet-18, ResNet-50, ResNet-101, and ResNet-152, respectively). ResNet is comparable with VGG-16 except that it has additional identity mapping capability (Theckedath & Sedamkar, 2020).

The Densely Connected Convolutional Network (DenseNet) was introduced and studied by Huang et al. (2018). It is easier to train a model through DenseNet because the network reduces the number of parameters and improves the gradient and information flow throughout the network. Because of this, DenseNet became popular in feature reuse where the output of each layer to another layer is being connected, which makes models be easily trained and parameter efficient.

Since it is difficult to collect and expensive to train data, we utilize transfer learning, which essentially utilizes knowledge (feature weights) acquired for one task to solve related ones (Weiss et al., 2016). In transfer learning, the usual approach is to train a base network and then copy its first n layers to the first n layers of the new network (Yosinski et al., 2014). A fine-tuning method, common to radiology research, is to backpropagate the errors from the new network into the base features. Alternatively, the features of the base network can be frozen while fine-tuning the rest of the deep layers (Yamashita et al., 2018).

Data Preprocessing

Before training the model, the images were resized to 224 × 224 and each channel of tensor was normalized with a mean of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225) so the pixels would range from 0 to 1 in order to help the model converge during the training phase.

In addition, we employed data augmentation such as center cropping and random horizontal flipping to artificially increase the size and quality of the training domain.

Training and Implementation Details

There are two major experimental setups in training the CNN models: (1) training with only the original train set (NO ACGAN) and (2) training with the original train set + ACGAN-generated images (WITH ACGAN).

Since the child pneumonia x-ray public data set in Kaggle is already partitioned into train, test, and validation sets, with 5,216, 624, and 16 instances, respectively, we simply utilize the suggested train set for the first experimental setup. The distribution of the data set is specified in Table 1. Meanwhile, for the second setup, we combine the original training set with the synthetic images generated by ACGAN to increase the amount of the training samples. The number of ACGAN-generated synthetic images combined with the original training data is of value $k = 500$ or $k = 1{,}000$ per class. These are randomly selected samples from a total of 2,000 synthetic images produced by ACGAN per class. We evaluate the performance of the two in order for us to determine the optimal number of synthetic images.

For faster training, we use the version of CNN models that have been pretrained on the ImageNet data set. Then, 4,096-dimensional features from the last fully connected layer were extracted for the VGG-16 while 1,024 were extracted for ResNet and DenseNet. A fully connected layer with a dimension of 2 is added since there are two classes involved (pneumonia or normal).

For each experimental setup, we searched for the optimal hyperparameter configuration as shown in Table 4. We performed grid search, where we tried every possible configuration of the parameters. Essentially, we define a grid on n dimensions, where each cell in the grid maps to a pair of hyperparameters, for example, n = (learning rate, batch size). We define the range of possible values for batch size = [16, 32] and learning rate = [0.001, 0.0001]. Additionally, for each pair (learning rate, batch size), we also perform grid search on the number k of random ACGAN-generated synthetic images per class to be added to the original training set, that is, k = [500, 1000]. For each configuration, we repeat the process of training and testing on all the CNN models.

Table 4. Grid Search for Optimal Hyperparameter Configuration

Number of ACGAN Synthetic Images Added (per Class)	Batch Size 32		Batch Size 16	
	Lr = 0.001	Lr = 0.0001	Lr = 0.001	Lr = 0.0001
500	Implemented	Implemented	Implemented	Implemented
1000	Implemented	Implemented	Implemented	Implemented

Additionally, we use Adam, a method for stochastic optimization that calculates adaptive learning rates for parameters (Waheed et al., 2020) and cross-entropy loss function as the optimizer. We train the networks for 50 epochs, but we perform predictions on the test set using the model that has achieved the best validation accuracy.

Some studies have shown that increasing larger batches to improve the training time consequently causes performance degradation on the CNNs (Kandel & Castelli, 2020). Moreover, since we performed the experiments on Google Colab due to the lack of hardware availability to perform our experiments, the graphics processing unit (GPU) allocation is dynamic and limited. Using bigger batches than 64 leads to out-of-memory error, which limits our hyperparameter tuning to batches 16 and 32.

Evaluation Metrics

To evaluate the classification performance of the CNN models, we calculate the precision, recall, accuracy, and F_1-score, whose equations are given below:

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

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

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

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where

TP (True Positive)—the number of pneumonia samples that are correctly classified
TN (True Negative)—the number of normal samples that are correctly classified
FP (False Positive)—the number of normal samples that are wrongly classified
FN (False Negative)—the number of pneumonia samples that are wrongly classified

Tools

The experiments were conducted using Python and PyTorch in the Google Colaboratory environment. Other python packages such as scikit-learn, numpy, time (to measure the training execution), and matplotlib were also used to implement some parts of the model.

However, due to Google Colab's dynamic allocation of GPU, the time of execution might not be accurate and might differ across experiments.

RESULTS AND DISCUSSION

In this section, we present the results of our proposed method, which uses ACGAN for data augmentation and pretrained CNNs for the detection of pneumonia in chest x-ray. In summary, we conducted the following experiments:

1. Compare the performance of different classic CNN models, particularly VGG-16, ResNet, and DenseNet-12,

in detecting pneumonia using only the original training set.

2. Compare the performance of models in (1) detecting pneumonia using the original training set + ACGAN-generated samples.

We performed grid search to determine the best hyperparameter configuration as shown in Table 4. Since we have a total of eight configurations (= *learning rate* × *batch size* × *k*), we only report and discuss the best results and configurations, which are displayed in Table 5.

Table 5. CNN Best Parameters

Parameter	Value
Batch size	16
Learning rate	0.0001
Maximum epochs	50
Criterion	Cross-entropy
Optimization	Adam

GAN Results

As shown in Table 3, the ACGAN model was trained up to 600 epochs. In Figure 4, we compare samples from the original training set (a) with the images generated by ACGAN (c). The structure of lungs and white opacities are already learned by GAN and can already be observed in the 100th epoch. However, the image is still hampered by a lot of noise. The image generated slightly improved in the 200th epoch with less white opacities and noise clouding the image. The images generated began to worsen on the 300th epoch up to the 600th epoch, which means that the model started to overfit and learn the irrelevant details.

In order to see the changes clearly as we progress from one epoch to another, we thresholded the images into a binary image in order to separate the lung structure from the white opacities. As we can see on the thresholded version of the original images (b), the edges and the rib structure pointed out by the yellow arrows, represented by the black regions, are clearly visible. When we compare it to that of the ACGAN-generated thresholded images (d), we can see the structure of the lungs forming in the 100th epoch and slightly larger and more discernible lungs in 200th epoch but begin losing information in the 300th epoch. The 300th epoch had barely formed any structure since the whole image is covered by white opacities. The same is true with images produced in the 400th and 500th epochs.

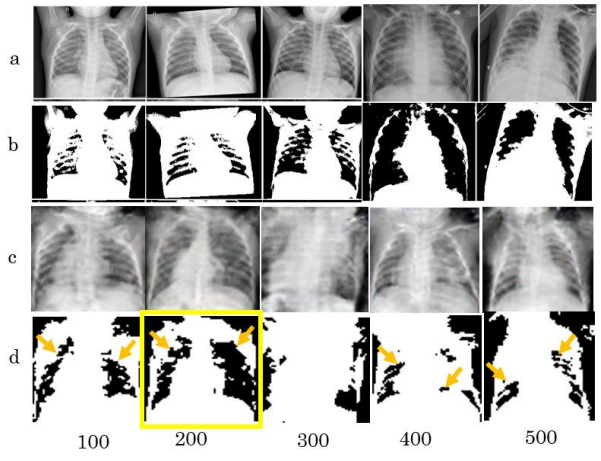

Figure 4. (a) Sample images from the original training set, (b) thresholded image of a, (c) ACGAN-generated synthetic images in every 100th epoch, and (d) thresholded image of d.

In Figure 5, we can observe samples of randomly generated normal and pneumonia chest x-ray images in the 220th epoch, which particularly showed to have the best set of images. As we can see in the same figure, the images have a clear lung structure similar to the features of the original samples.

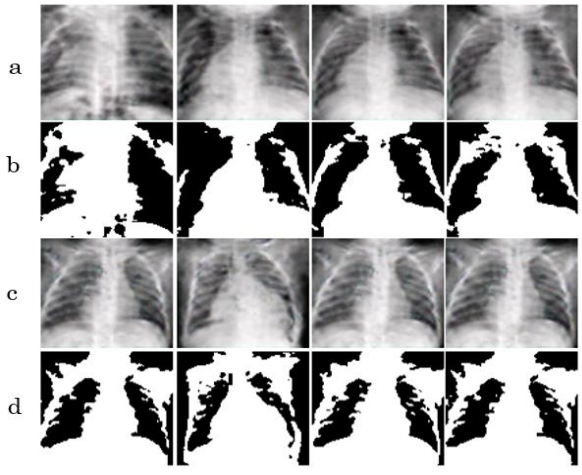

Figure 5. ACGAN-generated images on the 220th epoch. (a) Synthetic images labeled with pneumonia, (b) thresholded version of a, (c) synthetic images labeled as normal, and (d) thresholded version of c.

Performance of Pretrained CNNs (No ACGAN)

Table 6 displays the results of training the CNN models with the parameter configurations displayed in Table 5. As observed, ResNet-152 has outperformed the other models with 85.58% accuracy. It is closely followed by DenseNet-121, the accuracy of which is slightly lower than that of ResNet-152 by 3.21%. Similar observations can be noted for other metrics such as precision, recall, and F_1-score.

For VGG-16 and the other ResNet variants such as ResNet-18, ResNet-50, and ResNet-101, the accuracy only ranged between 70% and 79%. Based on these observations, we note that the CNN models with a higher number of layers, that is, ResNet-152 and DenseNet-121, are more effective in detecting pneumonia in chest x-ray images.

Table 6. Performance of Pretrained CNNs

	Precision	**Recall**	**F_1-Score**	**Accuracy**	**Training Time**
VGG-16	0.7937	0.7067	0.6418	0.7067	154m 52s
DenseNet-121	0.8602	0.8237	0.8080	0.8237	77m 32s
ResNet-18	0.8303	0.7789	0.7507	0.7788	73m 37s
ResNet-50	0.8341	0.7853	0.7593	0.7853	76m 53s
ResNet-101	0.8137	0.7436	0.6998	0.7436	200m 8s
ResNet-152	**0.8763**	**0.8558**	**0.8473**	**0.8558**	**121m 16s**

CNN With and Without ACGAN

In this section, we present the performance of CNN models trained with the combined original training set and synthetic samples generated by ACGAN, as shown in Table 7. Additionally, we present a performance comparison of CNN models with and without data augmentation with ACGAN in Figure 6. Note that these are results from using the parameters in Table 5.

As we can see in Figure 6, all the CNN models have displayed an improvement in all the metrics for data augmentation using the synthetic images generated by ACGAN. ResNet-18 has demonstrated a superior performance with its 91.24% accuracy. In addition, ResNet-18 was shown to have the biggest improvement with an increase of 13.36% in accuracy and 16.13% in F_1-score when trained in conjunction with ACGAN-generated images.

Table 7. Performance With ACGAN

	Precision	Recall	F_1-Score	Accuracy	Training Time
VGG-16	0.7944	0.7303	0.7012	0.7303	42m 59s
DenseNet-121	0.8662	0.8292	0.8201	0.8292	36m 24s
ResNet-18	**0.9128**	**0.9124**	**0.9120**	**0.9124**	**25m 14s**
ResNet-50	0.902	0.8989	0.8979	0.8989	29m 35s
ResNet-101	0.8852	0.8652	0.8608	0.8652	43m 39s
ResNet-152	0.8491	0.8180	0.8089	0.8180	60m 1s

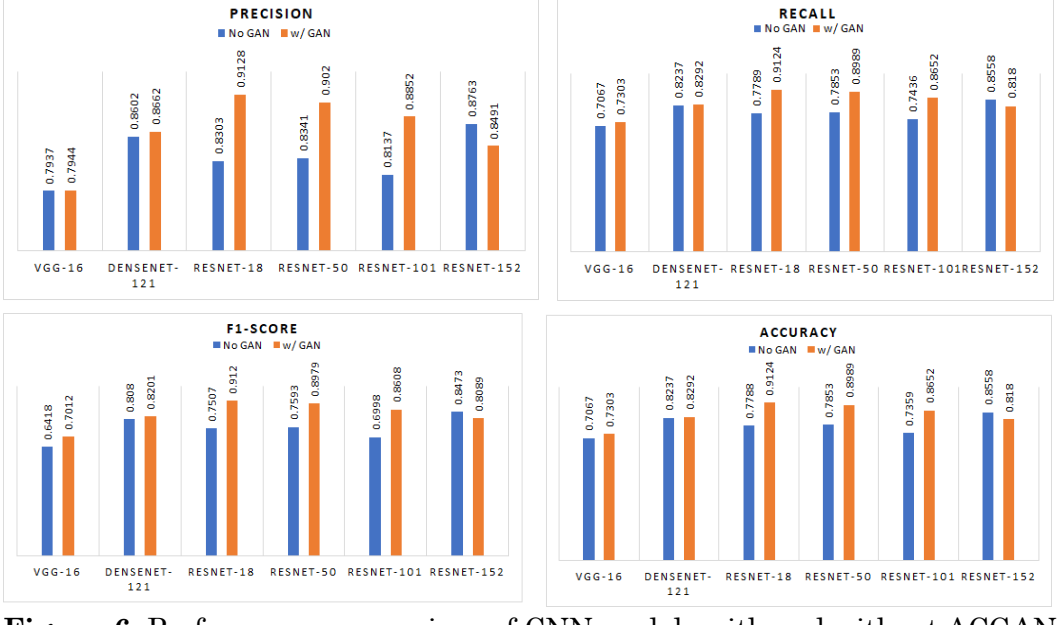

Figure 6. Performance comparison of CNN models with and without ACGAN.

In our experiment, the images generated by ACGAN in Figure 5 display a similarity with the realistic samples in Figure 3. The synthetic images generated were of low quality compared to the real samples; however, more realistic images were generated from the 220th epoch. Thus, by adding these samples to the training samples, the variability in the training samples has been increased, which is apparent in the improved classification results displayed in Table 8.

In addition, as mentioned, we experimented on the impact of varying the number of synthetic images that have been combined with the original training sets. We set the number of ACGAN-generated images to either k = 500 or k = 1,000 per class (e.g., 500 pneumonia, 500 normal). We can observe the results of this experiment in Table 8, where we perform a comparison in terms of accuracy and training time. As we can see, only VGG-16 and ResNet-152 are the only models where k = 1,000 produced better accuracy than k = 500. Despite this, ResNet-18 using k = 500 is still superior by 6%.

Table 8. Performance With ACGAN With Varying Number of Synthetic Images **(*k*)**

	Accuracy		**Training Time**	
	***k* = 500**	***k* = 1,000**	***k* = 500**	***k* = 1,000**
VGG-16	0.7303	0.7420	42m 59s	148m 59s
DenseNet-121	0.8292	0.8141	36m 24s	96m 47s
ResNet-18	**0.9124**	0.8573	**25m 14s**	67m 46s
ResNet-50	0.8989	0.777	29m 35s	88m 6s
ResNet-101	0.8652	0.8301	43m 39s	141m 14s
ResNet-152	0.8180	0.8510	60m 1s	162m 55s

These have been achieved with the CNN parameters: batch_size = 16, learning_rate = 0.0001, epochs = 50, and n_images = 50 and the ACGAN parameters batch size = 32, lr = 0.002, epochs = 220, and max_epoch = 600.

Moreover, using k = 1,000 per class, the data set size is increased by 2/3 of the original size of the training set (5,216). This consequently lengthened the time required to train the models, which is not ideal in settings where we want to maximize the number of experiments. With this, we have used k = 500 for the rest of our experiments.

The performance of ResNet variants such as ResNet-50 and ResNet-101 has also been amplified with ACGAN, as observed with their 11.36% and 12.93% increase in accuracy. On the other hand, VGG-16 and DenseNet-121 only displayed minimal improvement in accuracy with 2.36% and 0.55% increase, respectively. The effectiveness of ResNet may be attributed to the skip-connections embedded in its architecture that help overcome the issue of vanishing gradients.

What is surprising is the performance of ResNet-152 in this setup. Contrary to the results of the CNN models we have presented in the previous section, ResNet-152 has been outperformed by the other ResNet variants and DenseNet-121, with only 81.80% accuracy. Adding the synthetic images generated by ACGAN, its accuracy has dropped by 3.78%. The accuracy drop of ResNet-152 with the addition of GAN-generated synthetic images may be due to the network memorizing the training set due to overfitting. Overfitting is the state by which the network learns the training set that it started to model the noise in the training samples. Since ResNet-152 is the largest network, it has the capacity to learn more features, and it learns the noise/incorrect information from the synthetic images.

Also, since we have standardized our parameters, across experiments, the new hyperparameters might not have worked properly with the new set of data, and no new regularization techniques were embedded across the CNN models.

We have shown the superiority of ResNet in dealing with this specific task. Also, with ResNet-18, the number of parameters required to achieve good results has been reduced and consequently reduced the cost of training time, as compared to other models. This claim is supported by the superior performance of ResNet-18 despite having fewer layers than ResNet-152 and other CNN models. As we can see in Table 5, ResNet-18 completed its training after 25 minutes and 14 seconds, which has cut down the training time of ResNet-152, which showed the best performance without ACGAN, by more than double. Cutting down the training time allows us to perform more experiments and hyperparameter tuning. Moreover, with ResNet-18 having fewer layers than ResNet-152, we have the advantage of using lower hardware requirements, which allows us to develop cost-effective devices. Lower hardware requirements also enable the use of increased image resolutions.

With these observations, data augmentation with ACGAN can effectively reduce the number of trainable parameters required to achieve a relatively good performance in detecting pneumonia in chest x-ray images. Indeed, the addition of ACGAN-generated images as additional samples has added variability to the data set, hence reducing overfitting and improving the results on the test set.

CONCLUSION

In this study, we were able to detect pneumonia from a collection of chest x-ray images using pretrained classic CNN architectures such as VGG-16, DenseNet-121, ResNet-152, ResNet-18, and ResNet-50. In order to overcome the limited data availability and to improve the performance

of these models, we employ data augmentation by producing realistic synthetic samples with auxiliary classifier GAN (ACGAN) that are combined with the original training set. We performed grid search on parameters such as batch size, learning rates, number of epochs, and number of synthetic images, to determine the optimal hyperparameter configuration that will be standardized across the experiments and therefore allow us to have a fair comparison.

Our results show that CNN architectures with a higher number of layers such as ResNet-152 and DenseNet-121 are more effective in detecting pneumonia from the images. Training with ACGAN on the other hand has produced a different and surprising result. ResNet-18, the smallest out of all the ResNet variants, has outperformed all the models and recorded the highest improvement of 13.36% accuracy and 16.13% in F_1-score when trained in conjunction with ACGAN-generated images. Mid-sized networks such as ResNet-50 and ResNet-100 also outperformed the other models such as DenseNet-121 and VGG-16.

Based on these observations, it was shown that ResNet had an excellent performance compared to other models in the study. Specifically, ResNet-18 had the best performance and effectively cut down the training time. We have shown that synthetic images generated by ACGAN can introduce variability to the data set that helps reduce overfitting. Also, all models except ResNet-152 have improved, which means that ACGAN is worthy of investigating whether it can further improve other existing models for the detection of pneumonia.

As early detection and diagnosis of pneumonia is a vital point to save lives, accurate prediction is required, and we have shown that the use of ACGAN to produce synthetic images can increase the probability of having higher prediction accuracy. However, we note that these synthetic samples produced are not intended to replace the actual chest x-ray images but rather just add variability to the training set. There are still a bunch of limitations that we aim to address in the future. No quantitative metrics to measure the distance of similarity between the generated images and samples have been considered. We only based on human judgment to determine whether the images are clear or void of noisy artifacts. More so, no radiologists have been consulted on whether the generated results are indeed similar to the original samples.

In the future, we aim to further tweak the architecture of CNN models or explore other CNN models while in conjunction with the proposed ACGAN method for data augmentation. There is also a need for more experiments on different tasks and data sets to determine if ACGAN can still produce an excellent performance in terms of accuracy especially with low-resolution images.

REFERENCES

Albert, B. (2020). Deep learning from limited training data: Novel segmentation and ensemble algorithms applied to automatic melanoma diagnosis. *IEEE Access, 8*, 31254–31269. https://doi.org/10.1109/ACCESS.2020.2973188

Dong, X., Lei, Y., Tian, S., Wang, T., Patel, P., Curran, W. J., Jani, A. B., Liu, T., & Yang, X. (2019, December). Synthetic MRI-aided multi-organ segmentation on male pelvic CT using cycle consistent deep attention network. *Radiotheraphy and Oncology*, *141*, 192–199. https://doi.org/10.1016/j.radonc.2019.09.028

Fetty, L., Lofstedt, T., Heilemann, G., Furtado, H., Nesvacil, N., Nyholm, T., Georg, D., & Kuess, P. (2020, May). Investigating conditional GAN performance with different generator architectures, an ensemble model, and different MR scanners for MR-SCT conversion. *Physics in Medicine & Biology, 65*(10), 105004. https://doi.org/10.1088/1361-6560/ab857b

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspam, H. (2018, January). *Synthetic data augmentation using GAN for improved liver lesion classification*. arXiv.org. https://arxiv.org/abs/1801.02385

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020, November). Generative adversarial networks. *Communications of the ACM, 63*(11), 139–144. https://doi.org/10.1145/3422622

Han, C., Kitamura, Y., Kudo, A., Ichinose, A., Rundo, L., Furukawa, Y., Umemoto, K., Li, Y., & Nakayama, H. (2019). Synthesizing diverse lung nodules wherever massively: 3D multi-conditional GAN-based CT image augmentation for object detection. In *2019 International Conference on 3D Vision (3DV)* (pp. 729–737). IEEE. https://doi.org/10.1109/3DV.2019.00085

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition*. arXiv.org. https://arvix.org/abs/1512.03385

Huang, G., Liu, Z., Maaten, L. V., & Weinberger, K. Q. (2018, January 28). *Densely connected convolutional networks*. arXiv.org. https://arxiv.org/abs/1608.06993

Jain, R., Nagrath, P., Kataria, G., Kaushik, V. S., & Hemanth, D. J. (2020). Pneumonia detection in chest x-ray images using convolutional neural networks and transfer learning. *Measurement*, *165,* 108046. https://doi.org/10.1016/j.measurement.2020.108046

Jmour, N., Zyen, S., & Abdelkrim, A. (2018). Convolutional neural networks for image classification. In *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)* (pp. 397–402). IEEE. https://doi.org/10.1109/ASET.2018.8379889

Kan, C. N. E., Maheenaboobacker, N., & Ye, D. H. (2020). Age-conditioned synthesis of pediatric computed tomography with auxiliary classifier generative adversarial networks. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)* (pp. 109–112). IEEE. https://doi.org/10.1109/ISBI45749.2020.9098623

Kandel, I., & Castelli, M. (2020, December). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, *6*(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Kermany, D., Zhang, K., & Goldbaum, M. (2018). *Labeled optical coherence tomography (OCT) and chest x-ray images for classification* (Version 2) [Data set]. Mendeley Data. https://doi.org/10.17632/rscbjbr9sj.2

Kim, J., Kim, J., Han G., Rim, C., & Jo, H. (2020). Low-dose CT image restoration using generative adversarial networks. *Informatics in Medicine Unlocked, 21*, 100468. https://doi.org/10.1016/j.imu.2020.100468

Learn about pneumonia. (n.d.). https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/learn-about-pneumonia#:~:text=Pneumonia%20is%20an%20infection%20of,to%20get%20into%20your%20bloodstream

Li, Y., Zhang, Z., Dai, C., Dong, Q., & Badrigilan, S. (2020). Accuracy of deep learning for automated detection of pneumonia using chest x-ray images: A systematic review and meta-analysis. *Computers in Biology and Medicine,*

123, 103898. https://doi.org/10.1016/j.compbiomed.2020.103898

Liang, G., & Zheng, L. (2020). A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Computer Methods and Programs in Biomedicine, 187*, 104964. https://doi.org/10.1016/j.cmpb.2019.06.023

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2794–2802). IEEE. https://doi.org/10.1109/iccv.2017.304

Mirza, M., & Osindero, S. (2014, November). *Conditional generative adversarial nets*. arXiv.org. https://arxiv.org/abs/1411.1784

Odena, A., Olah, C., & Shlens, J. (2017). Conditional image synthesis with auxiliary classifier GANs. *Proceedings of the 34th International Conference on Machine Learning*. *PMLR*, *70*, 2642–2651.

Pneumonia. (n.d.). https://www.who.int/news-room/fact-sheets/detail/pneumonia

Qian, P., Xu, K., Wang, T., Zheng, Q., Yang, H., Baydoun, A., Zhu, J., Traughber, B., & Muzic, R. F. (2020, June). Estimating CT from MR abdominal images using novel generative adversarial networks. *Journal of Grid Computing, 18*(2), 211–226. https://doi.org/10.1007/s10723-020-09513-3

Qin, C., Yao, D., Shi, Y., & Song, Z. (2018, August 22). Computer-aided detection in chest radiography based on artificial intelligence: A survey. *BioMedical Engineering OnLine*, *17*, Article 113. https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0544-y

Radford, A., Metz, L., & Chintala, S. (2016, January). *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv.org. https://arvix.org/abs/1511.06434

Saraiva, A., Ferreira, N., de Sousa, L. L., Costa, N., Sousa, J., Santos, D., Valente, A., & Soares, S. (2019). Classification of images of childhood pneumonia using convolutional neural networks. In *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2019)* (112–119). https://www.scitepress.org/Link.aspx?doi=10.5220/0007404301120119

Shaikhina, T., & Khovanova, N. A. (2017, Jan). Handling limited datasets with neural networks in medical applications: A small-data approach. *Artificial Intelligence in Medicine, 75*, 51–63. https://doi.org/10.1016/j.artmed.2016.12.003

Theckedath, D., & Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 Networks. *SN Computer Science*, *1,* 79. https://doi.org/10.1007/s42979-020-0114-9

Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., & Pinheiro, P. R. (2020). CovidGAN: Data augmentation using auxiliary classifier GAN for improved Covid-19 detection. *IEEE Access, 8*, 91916–91923. https://doi.org/10.1109/ACCESS.2020.2994762

Wani, M. A., Bhat, F. A., Afzal, S., & Khan, A. I. (2020). *Advances in deep learning*. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-13-6794-6

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*, 9. https://doi.org/10.1186/s40537-016-0043-6

Wolterink, J. M., Leiner, T., Viergever, M. A., & Isgum, I. (2017, December). Generative adversarial networks for noise reduction in low-dose CT. *IEEE Transactions on Medical Imaging, 36*(12), 2536–2545. https://doi.org/10.1109/TMI.2017.2708987

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights Into Imaging*, *9*(4), 611–629.

Yang, H., Xia, K., Anqi, B., Qian, P., & Khosravi, M. R. (2019). Abdomen MRI synthesis based on conditional GAN. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1021–1025). IEEE. https://doi.org/10.1109/CSCI49370.2019.00195

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). *How transferable are features in deep neural networks?* arXiv.org. https://arxiv.org/abs/1411.1792