Original Article

Generative Adversarial Network Image Synthesis Method for Skin Lesion Generation and Classification

Abstract

Background: One of the common limitations in the treatment of cancer is in the early detection of this disease. The customary medical practice of cancer examination is a visual examination by the dermatologist followed by an invasive biopsy. Nonetheless, this symptomatic approach is time-consuming and prone to human errors. An automated machine learning model is essential to capacitate fast diagnoses and early treatment. Objective: The key objective of this study is to establish a fully automatic model that helps Dermatologists in skin cancer handling process in a way that could improve skin lesion classification accuracy. **Method:** The work is conducted following an implementation of a Deep Convolutional Generative Adversarial Network (DCGAN) using the Python-based deep learning library Keras. We incorporated effective image filtering and enhancement algorithms such as bilateral filter to enhance feature detection and extraction during training. The Deep Convolutional Generative Adversarial Network (DCGAN) needed slightly more fine-tuning to ripe a better return. Hyperparameter optimization was utilized for selecting the best-performed hyperparameter combinations and several network hyperparameters. In this work, we decreased the learning rate from the default 0.001 to 0.0002, and the momentum for Adam optimization algorithm from 0.9 to 0.5, in trying to reduce the instability issues related to GAN models and at each iteration the weights of the discriminative and generative network were updated to balance the loss between them. We endeavour to address a binary classification which predicts two classes present in our dataset, namely benign and malignant. More so, some well-known metrics such as the receiver operating characteristic -area under the curve and confusion matrix were incorporated for evaluating the results and classification accuracy. **Results:** The model generated very conceivable lesions during the early stages of the experiment and we could easily visualise a smooth transition in resolution along the way. Thus, we have achieved an overall test accuracy of 93.5% after fine-tuning most parameters of our network. **Conclusion:** This classification model provides spatial intelligence that could be useful in the future for cancer risk prediction. Unfortunately, it is difficult to generate high quality images that are much like the synthetic real samples and to compare different classification methods given the fact that some methods use non-public datasets for training.

Freedom Mutepfe[1], **Behnam Kiani Kalejahi**[1,2], **Saeed Meshgini**[2], **Sebelan Danishvar**[3]

[1]Department of Computer Science and Engineering, School of Science and Engineering, Khazar University, Baku, Azerbaijan, [2]Department of Biomedical Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran, [3]Department of Electronic and Computer Engineering, Brunel University, London, UK

Keywords: *DCGAN, dermoscopy, pretraining, skin lesion*

Submitted: 17-Jul-20 **Revised:** 29-Sep-20 **Accepted:** 01-Jan-2021 **Published:** 20-Oct-2021

Introduction

Skin cancer is a common health challenge around the globe.[1] Early screening of skin cancer is of paramount importance to curb mortality and increase the possibilities of survival rate in patients. Skin lesion analysis plays an essential role in skin cancer prophylaxis, notably in terms of getting an effective early diagnosis. Having learned the integral concepts of neural networks in the previous semester, the researcher now ought to augment the practical skills out of the sphere of this module by replicating a research article on unsupervised deep learning. Even though an increasing amount of data is becoming available on the Internet, the vast majority of it remains unlabeled. In this context, we leveraged the ready unlimited number of unlabeled skin cancer images to learn proper machine learning techniques, which can then be utilized on number of unsupervised learning tasks like image classification.[2] Generative models are one of the most prevalent methodologies that are applied on classification problems. The model has to examine and understand the gist of the training image data before it can generate comparable results itself.[3]"What I cannot create, I do not understand", stated

For reprints contact: WKHLRPMedknow_reprints@wolterskluwer.com

How to cite this article: Mutepfe F, Kalejahi BK, Meshgini S, Danishvar S. Generative adversarial network image synthesis method for skin lesion generation and classification. J Med Sign Sens 2021;11:237-52.

Address for correspondence:
Dr. Saeed Meshgini,
Department of Biomedical Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran.
E-mail: smeshgini@gmail.com

Access this article online

Website: www.jmssjournal.net

DOI: 10.4103/jmss.JMSS_53_20

Quick Response Code:

the renowned physicist Richard Feynman. It can be said that, for models to understand their input, they need to learn to create similar data samples to explore this quote in the circumstance of machine learning. The most promising approach is to use the generative models that learn to discover the essence of data and find the best distribution to represent it. Generative Adversarial Network (GAN)[3] is contemplated as a principal group of models to build satisfactory image depictions and generate " realistic " images. The notion of GAN is to synchronously train two distinct models, namely the generative model G and the discriminative model D. The generative model produces convincing images that are similar to a real-image data distribution while the assignment of the discriminative model is to ascertain whether or not a considered image looks real or not. To understand a generative modeling technique, it may be comparable to a criminal gang of counterfeiters, attempting to fabricate counterfeit currency and make use of it without getting caught, whereas the discriminative model is similar to law enforcement agents, endeavor to uncover the counterfeit money. In this context, contention between the two helps them to enhancing their tactics up to a time where law enforcement agents would not be able to discern the counterfeits from real currency. In this study, our objective is to investigate a remarkable situation where a multi-layered generative network produces examples that capture the distribution of our original image dataset and an equally multi-layered perceptron called discriminator endeavor to make a distinction between fake and real image samples. In this instance, we will be able to train these models mutually employing only the enormously successful dropout and backpropagation algorithms[4] as well as an instance that could have come from the generative model after applying only forward propagation. To assist dermatologists in making their diagnostic study, this work seeks to propose an improved cutaneous lesions scrutiny and classification procedure employing DCGAN and assess a set of constraints as well as hyperparameters on this architectural topographic anatomy.

In accession toward this image segmentation approach, dermal lesions are processed by applying effective image filtering and enhancement algorithms to enhance the model feature detection and extraction during training. This study is structured in the following way. We have examined some related work in the realm of skin cancer classification in section 2. In addition, we also have discussed about the previous knowledge of GANs, their advancement and constraints in section 3. In section 4, we demonstrated our experimental approach. Furthermore, Section 5 describes the dataset incorporated and two proposed image processing techniques. Section 6 then combines the results from the training stages and further analysis. Finally, Section 6 draws conclusions of the study and provides guidance for future research.

Related Work

Deep learning is an intuitive process whose complexity of learning increases with the increase in the number of layers. Due to its high performance, it is regarded as a mature application for medical diagnostics.[5] In recent times, deep learning has contributed significantly for skin lesion classification problems.[6] However, limited data set creates tougher environment for the potential ground-breaking research in medical diagnostics with deep learning. One reason is dependency of the deep learning algorithm on training data size as it requires millions of parameters and profusion of labelled data to learn.[7] When deep learning model uses limited data to train, it uses large amount of its resources to train the model, creating overfitting issues. Overfitting issue refer to model's incapability to generalize on unseen data. A large number of researches have been done to overcome challenges imposed by limited data on the training of deep learning models. It includes techniques such as augmentation,[8] transfer learning,[26] and ensemble of classifiers.[9] The following sections provide an overview of existing techniques and related works done in field of skin lesion classification.

In the recent times, generative algorithms have involved variational autoencoders and networks which are able to map from image space to latent space and back, or autoregressive models, which take actions from the previous step as input to deliberate on the value of the next step. However, the application of adversarial training into generative modelling occasioned a considerable step toward a more powerful method of synthesizing new data. Goodfellow *et al*., 2014[10] introduced GANs as a system of two neural networks, a generator and a discriminator, opposing one another. The former synthesizes images that match the data distribution which are then classified by the latter as either true or false. As the discriminator gets better at distinguishing the authenticity of the images, the generator is forced to enhance itself to be able to fool the discriminator, thus, slowly learning the structure of the data that passes through the network. Initially, both of them will show low effectiveness, the images generated will be essentially noise and the loss of the discriminator will be high. As the training advances, the results will start to resemble the data until the discriminator can no longer recognize real from fake. Given this mechanism of image generation, the set of available data can be further expanded, making the design and training of generative models for data augmentation a plausible choice. Moreover, in practice, increases in accuracy have been seen in several learning systems.[8]

We have witnessed a great deal of work published in the realm of skin cancer classification using deep learning and computer vision techniques. These approaches make the extensive use of diverse methods including detection, classification and segmentation, and image processing using a variety of filters for instance Karabulut *et al.*[10] created an algorithm for the classification of melanoma by utilizing Support Vector Machine (SVM) and k-means clustering. In Esteva *et al.*[11] had a breakthrough on skin cancer classification when he used a pre-trained GoogleNet Inception v3 CNN model to classify 129,450 clinical skin cancer images including 3,374 dermatoscopic images[12] used a deep convolutional neural network to classify the clinical images of 12 skin diseases.[13] Developed a convolutional neural network with over 50 layers on ISBI 2016 challenge dataset for the classification of malignant melanoma. In 2017,[11] utilized a deep convolutional neural network to classify a binary class problem of dermoscopy images.[14] designed an algorithm utilizing a deep convolutional neural network approach together with SVMs for the classification of four distinctive categories of clinical skin cancer images.

Introduced in 2016 by Alec Radford, Luke Metz, and Soumith Chintala,[15] DCGAN marked one of the most important early innovations in GANs since the technique's inception two years earlier.[13] This was not the first time a group of researchers tried harnessing Convolutional Neural Networks (ConvNets) for use in GANs, but it was the first time they succeeded at incorporating ConvNets directly into a full-scale GAN model.

The use of ConvNets exacerbates many of the difficulties plaguing GAN training, including instability and gradient saturation. Indeed, these challenges proved so daunting that some researchers resorted to alternative approaches, such as the LAPGAN, which uses a cascade of convolutional networks within a Laplacian pyramid, with a separate ConvNet being trained at each level using the GAN framework.[16] If none of this makes sense to you, don't worry. Superseded by superior methods, LAPGAN has been largely relegated to the dustbin of history, so it is not important to understand its internals. Although inelegant, complex, and computationally taxing, LAPGAN yielded the highest-quality images to date at the time of its publication, with fourfold improvement over the original GAN (40% vs. 10% of generated images mistaken for real by human evaluators). As such, LAPGAN demonstrated the enormous potential of marrying GANs with ConvNets. With DCGAN, Radford and his collaborators introduced techniques and optimizations that allowed ConvNets to scale up to the full GAN framework without the need to modify the underlying GAN architecture and without reducing GAN to a subroutine of a more complex model framework, like LAPGAN. One of the key techniques Radford *et al.* used is batch normalization, which helps stabilize the training process by normalizing inputs at each layer where it is applied. This work is based upon the aforementioned approaches. In this work, we have addressed a binary skin cancer classification approach which endeavors to predict two classes, namely benign and malignant. We have taken the advantage of common metrics such as confusion metric for evaluating our results. Our work seeks to achieve an improved accuracy, precision, recall, F1 score and receiver operating characteristic-area under the curve (ROC-AUC) of 0.861 as compared to some previous sophisticated methods.

Background

Generative adversarial network

Generative paradigms have been the most prevalent methodologies that are implemented for these kinds of situations. The model should be competent enough to examine and comprehend the essence attributed to training data before it can generate similar results.[17] GANs have turn out to be one of the predominant developments of generative deep learning methodologies since it was established.[18] The generator needs to understand how to generate data in such a way that the discriminator won't be able to discern between fake and real. The discriminator network has the assignment of discerning produced images from true images. The fundamental structure of GANs incorporates two multi-layered networks, that trained concurrently, a generative model G sublimates random vector **z** adapted from preceding distribution $P_{(z)}$ into image data, furthermore a discriminative model D make an attempt to draw a distinction between true images obtained from training images *P* and simulated samples from the generator G. An instance of a GANs framework notably trained upon MINST dataset[19] as indicated in Figure 1.[20]

Such networks are tutored conflictingly, in the form of two-player minimax game, until none of them could make additional advancement against one another, either the generator turns to be pretty good that the discriminator may not easily distinguish between true and false. An illustration of the GAN objective function is depicted as follows:

$$\min_{G}\max_{D} V(D,G) = \min_{G}\max_{D}[E_{X\sim P_{data}(x)}[\log D(x)] + E_{z\sim p_{z}(z)}[\log(1-D(G(z)))] \quad (1)$$

x implies the factual training data, **z** signifies latent features aggregated upon the generator, furthermore, *G (z)* portrays the sample originating from the generator given a noise vector z. *D (x)* connotes the discriminator's approximation

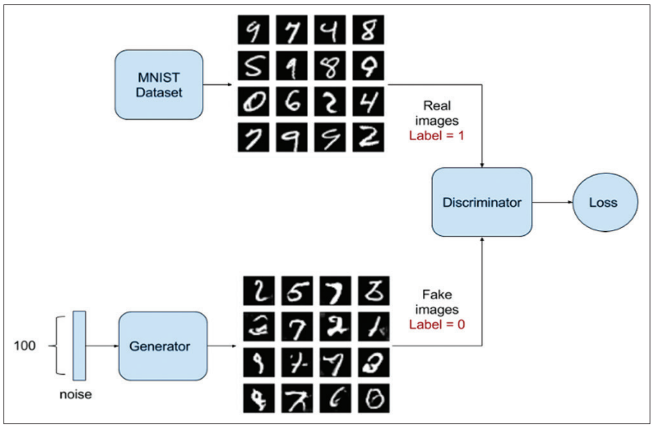

Figure 1: An illustration of the generative adversarial network training upon MNIST dataset. The generator attempts to produce images equivalent to images in MINST dataset thus the discriminator may not tell apart genuine images from generated images

```
for number of training recurrences do
for k steps do
    • sample minibatch of $m$ noise samples $\{z^{(l)},\ldots\ldots, z^{(m)}\}$ from noise prior $P_g(z)$
    • sample minibatch of $m$ examples $\{x^{(1)}\ldots, x^{(m)}\}$from data generating distribution $P_{data}{}^{(x).}$
    • Update the discriminator by ascending its stochastic gradient:

    $\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}[\log \mathrm{D}(x^{(i)}) + \log(1-D(G(z^{(i)})))]$

    end for

    • sample minibatch of $m$ noise samples $\{z^{(l)},\ldots\ldots, z^{(m)}\}$from noise prior $P_g(z)$
    • Update the generator by descending its stochastic gradient:

    $\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}\log(1-D(G(z^{(i)})))$

    end for
```

Figure 2: The algorithm of GAN here *k* is the considerable number of iterations carried out on the discriminator (hyperparameter)

of the possibility that real image data x is real, furthermore *D (x)* has to be as close as possible to 1, to perform better. The possibility that the discriminator determines if the samples obtained from the generator are true or false is represented by *D (G* (z*))*. Taking into account that the target of the generative model is to produce images analogous to the real images, the anticipation of the generator is for the value to be as big as possible. P_{data} *(x*). and P_Z *(z*) represents the probability density of x and z accordingly. In order to mislead Discriminator *D, G* is trained in a manner that diminishes log (1-D (G(z))). Contrarily *D* is trained so that it can enhance the likelihood that the generated data is authentic, illustrated by 1, bear in mind 0 constitute a fake. The narrative aforementioned is the underlying principle of the GAN is represented in Figure 2.[21] The algorithm for this model can be expected to take the following steps:

Algorithm

Minibatch stochastic gradient descent training of generative adversarial net. Here *k,* is a hyperparameter which represents number of steps to be implemented on the discriminator. We used k = 1 because it is a less expensive choice, in our experiment.

In the wake of consequences due to the contest of these two networks, GANs is well known to have limitations. Initially, there is an issue of mode collapse in the GANs.[22] Furthermore, due to high degree of autonomy, these networks exhibit some issues, such as nonconvergence and instability during execution. On the other hand, it is complex to train a GAN because it does not have a loss function, making it difficult throughout the learning process to determine if it has made some positive developments or not. To address those issues, Radford and counterparts[15] suggested a group of Convolution Networks named Deep Convolutional Generative Adversarial Networks (DCGAN), that possess a set of architectural constraints to balance GANs.[23] It is a set of broad lines for the establishment of architectures for images. This method is pretty common and in particular, this article has already been cited by many publications in accordance with Google Scholar.

Deep convolutional generative adversarial network

The principal idea of DCGAN is to broaden GAN using Convolution Network architectures. Radford[15] managed to attain stable results by endorsing certain architectural constraints to DCGAN.

Below principles were introduced in:[17]

- Modification of the generator by replacing pooling layers with stride convolutions in discriminator and fractional-strided convolutions
- The use LeakyReLU activation function in the discriminator over the entire layers
- Removal of fully connected layers on top of convolutional features and directly linking the outcome to the convolutional layers
- The use of ReLU activation function in the generator in all layers leaving out the output, which uses tanh activation function
- Batch normalization incorporated in either of the two, generator and discriminator.

The illustration of model architecture is expressed in Figure 3. Not entirely connected or pooling layers are implemented. The input z to the model is one hundred-dimensional vector habitually sampled from unvarying distribution.

METHOD

Details of DCGAN architecture

Deep convolutional GAN training process is built by the recurrence of the following efforts:

1. A collection of image data x is exploited to train network *D*, the discriminator
2. The generative network produces satisfactory image depictions or " realistic" images.

Eventually, the discriminator ***D*** is refreshed in accordance

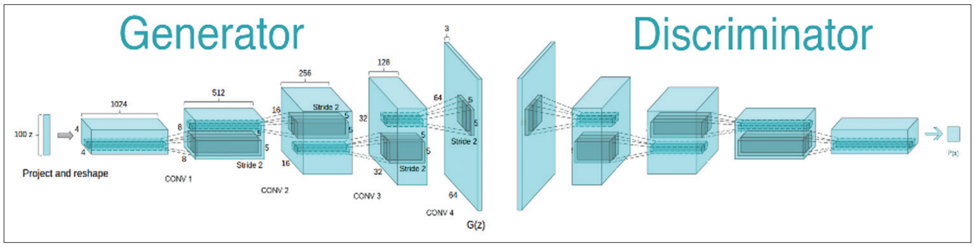

Figure 3: DCGAN framework employed on LSUN scene simulation[15]

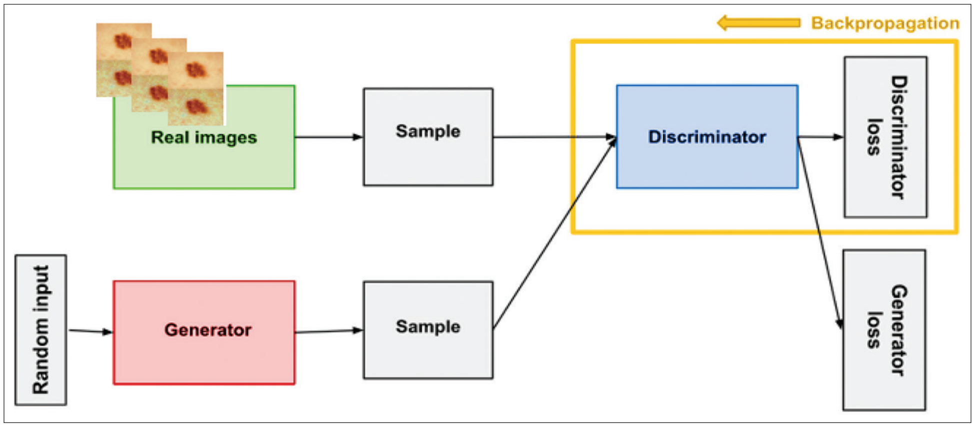

Figure 4: Illustration of backpropagation implemented on Discriminator *D*

with produced images. The goal of this procedure is for the generator ***G*** to produce images that are gradually more indistinguishable from the real images by incorporating a backpropagation and dropout algorithms[17] after repeated iterations, as Illustrated by Figure 4.

The design of DCGAN in this work is in reference to the code from a publicly accessible repository. This code was scripted to train a DCGAN model on the CIFAR-10 dataset.[25] We have amended this code to accomplish much resemblance as possible to the DCGAN framework stated within the preceding paragraph.

This necessitated building the appropriate code for our input size and modifying the number of appropriate filters in convolutional layers. The design of the trained model is indicated in Figure 3. We have ensured that we employ deconvolutions in generator and stridden convolutions in discriminator. Notably, Batch Normalization was applied and ReLU/LeakyReLU activation functions were brought into play as suggested by guidelines.

In a similar case of the Deep Convolutional GAN article,[19] solely transformation process of the images was just resizing them to range of tanh activation function [−1, 1]. This design was trained using mini-batch size 128. In addition, Adam optimizing algorithm was incorporated applying 0.0002 as the learning rate and 0.2 as the momentum β_1. The incline of the leak in the LeakyReLU was fixed to 0.2. The entire weights were initialized by default settings. In the present case, the noise **z** was taken from a standard normal distribution. The realization of this undertaking is performed using Keras in python anaconda development application software. Experiments were performed on an IBM computer equipped with Windows 10 Home 64 Bit and 10th Generation Intel Core i7-10750H 6-Core Processor. In addition, NVIDIA GeForce RTX 2060 graphics with 6 GB of dedicated GDDR6 VRAM was also utilized. More so with 16 GB DDR4 2933MHz Dual-Channel Memory and 512GB NV Me SSD. Elapsed Time for NVIDIA GeForce RTX 2060 took an estimate of 1 h, 35 min, 16 s.

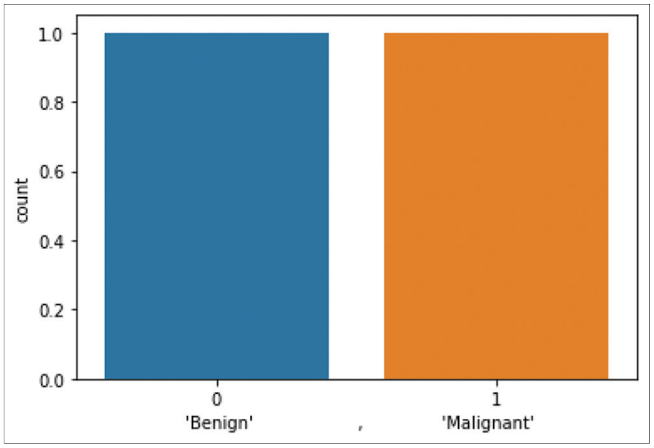

Figure 5: Illustration of data distribution in dataset

Data

We have trained our Deep Convolutional GANs (DCGAN) framework on skin cancer dataset 3,597 images from Kaggle,[26] which contain two classes namely benign and malignant. There is similar amount of skin lesion images in either of these classes as depicted in the Figure 5. In addition, before training our model, we incorporated some image processing techniques. The primary objective of processing images is usually to enhance the common image characteristics like picture quality and features and thereby suppressing undesired distortions, thereby making the result of our images appealing and clear for feature extraction and further examination.

Image processing

In this work, we have employed a bilateral filter to enhance our image data.[27] Commonly Gaussian Blur technique is exploited to reduce the amount of noise in an image and to get the desired texture as depicted in Figure 6. Nevertheless, this method presupposes pixels nearest to the middle pixel would have to be closest to the true value of that pixel, so they will sway the averaged value of the center pixel greater than pixels further away, which tend to blur edges. In this work, we want the edges for the benefit of the model to get the circumference of the lesion to perform better. We suggested the exploitation of a bilateral filter that is highly efficient at removing noise whilst conserving the edges.

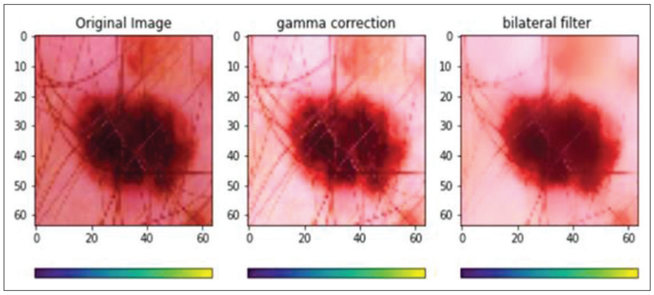

Figure 6: Example of images enhanced using gamma correction and bilateral filter. We further cropped the images into 64 × 64 pixels.

In addition, we have employed gamma-correction under the name of the power law transform[9] to transform our image data to the desired texture. Primarily, the intensities of our image pixels need to be adjusted from the pixel range of 0-255-0-1.0. Furthermore, to get the result of the corrected image, we implement succeeding formula:

$$O = I^{(1/G)} \tag{2}$$

In this formula, I represent our input image and G denotes the gamma value to be adjusted. In addition to that, O represents our resulting image and after gamma corrections it is then restored to the range 0–255.

In this instance, we tried $G = 1.0$ first, and then later implemented $G = 1.5$ and then our image data started to illuminate up and we witnessed more detail, which is enough value to attain a decent looking corrected image as the following Figure 7.

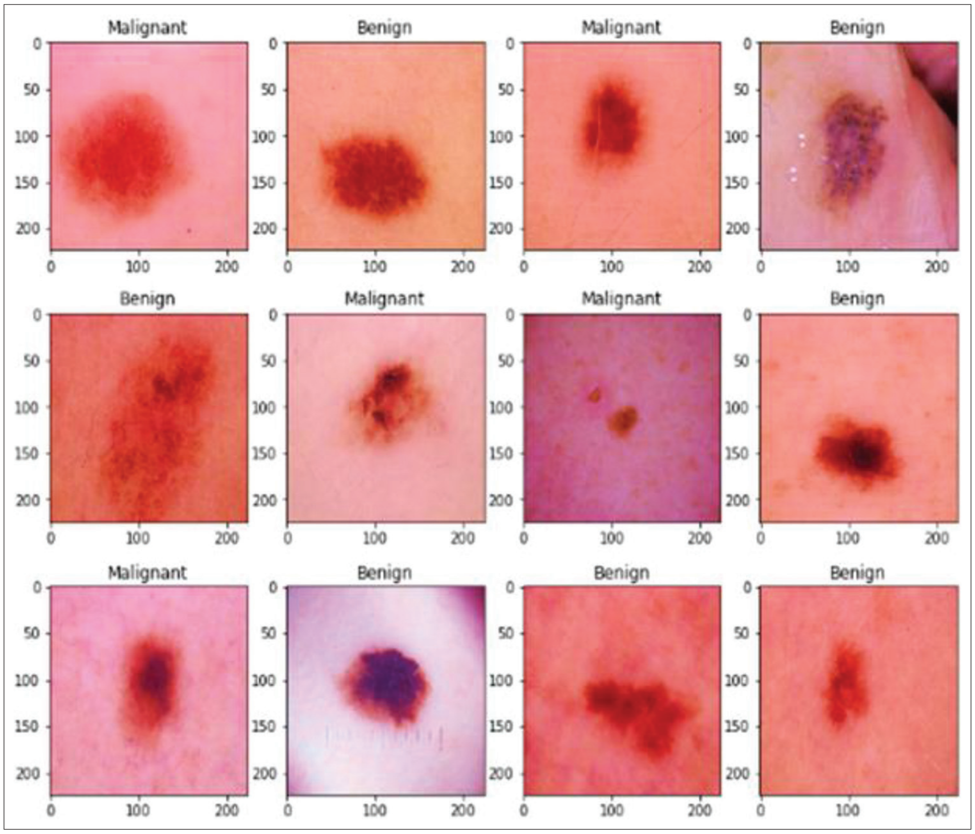

Figure 7: Example of images from the skin cancer dataset

Data augmentation

Considering that our dataset is not that big, data augmentation is quite handy to increase our dataset. Data augmentation is also one way to fight overfitting given the fact that our samples are most likely to be correlated when the dataset is small, which leads to overfitting. Overfitting[28] takes place when a model is exposed to a too-small set of data and becomes incapable to generalize on unseen data and to produce new set of data. Our principal objective for fighting overfitting is the entropic capacity of our model, for this reason, we incorporated this method to increase the size of our dataset by transforming existing images into a new form of the dataset using some transformation methods, such as rotation, shear, and flip as indicated in Figure 8.

Generator definition

In this work, the deconvolution neural network of our generator is put into operation by invoking conv2d_transpose method from the TensorFlow library to carry out weight multiplication as well as executing bias addition using the bias_add method. Furthermore, in order to carry out weight multiplication as well as executing bias addition, the

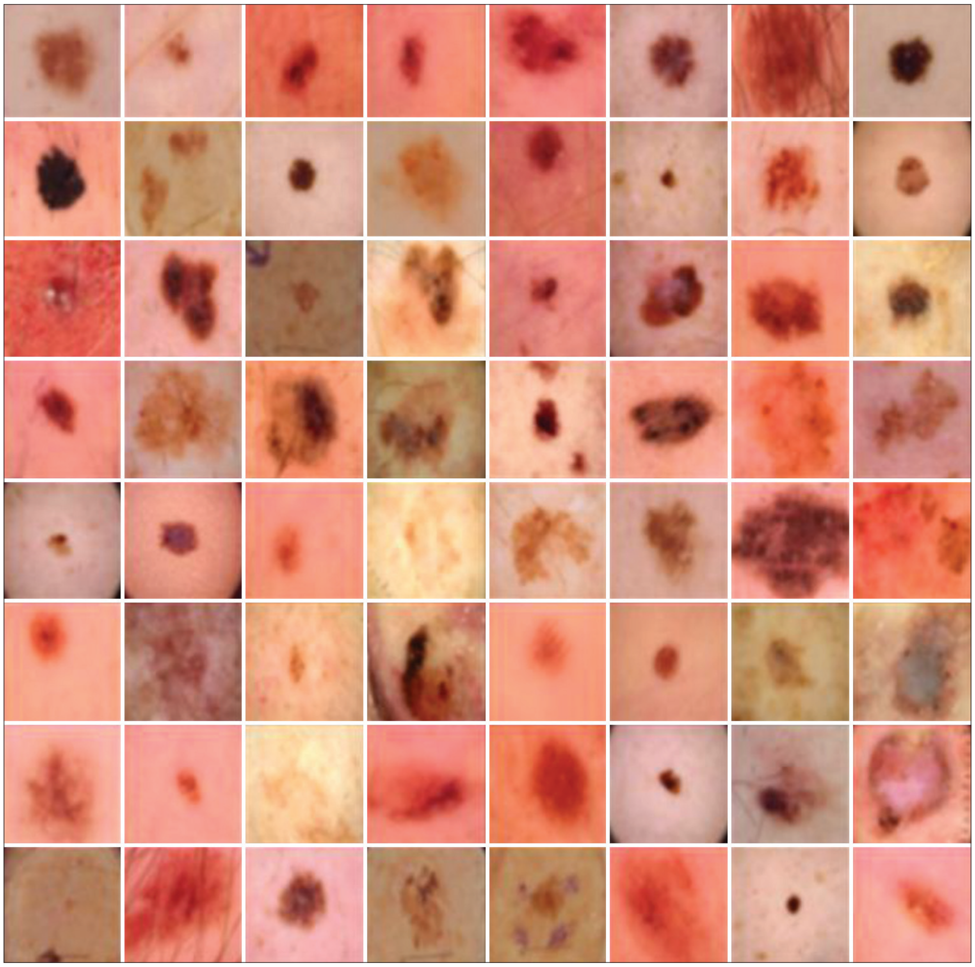

Figure 8: Example of images from skin cancer dataset after augmentation

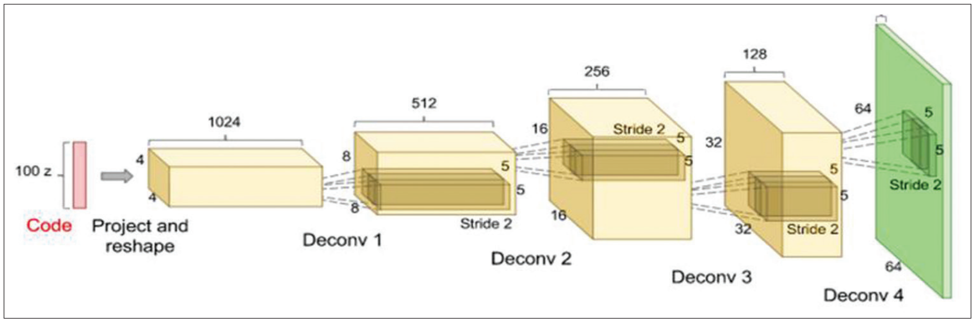

Figure 9: Generator, here the input is a random normal vector that passes through deconvolution stacks and outputs an image

convolution neural network in the discriminator also invoked the conv2d method in the TensorFlow library to accomplish these two functions. We constructed our generator addressing the needs of the DCGAN framework, additionally, we have fixed our OUTPUT_SIZE to 64 since our ultimate outcome is expected to be 64 × 64. The step of movement of the deconvolution is set to 2, and among all, each output augments fourfold than the input, so that we can get the output size of each layer were 32 × 32, 16 × 16.8 × 8 and 4 × 4 accordingly.

The BATCH_SIZE and GF is set to 64, respectively, additionally, the number of feature maps is set to 512, 256, 128, and 64 accordingly. Finally, the structure of the generator is portrayed in Figure 9.

Discriminator definition

The discriminator is a feed-forward neural network with five layers, including an input and an output layer, and three dense layers, and in this architecture, spatial pooling layers are absent. We plugged the input into the convolution layer. More so, the moving step of the convolution kernel is set to 2, furthermore, the output reduces to a quarter of the original ones after processing of convolution. With reference to that, the convolution layer output size becomes 32 × 32, 16 × 16, 8 × 8, 4 × 4, respectively, and also the number of feature maps was 64, 128, 256, and 512, respectively, as well. The final structure of the discriminator is illustrated in Figure 10.

Definition of training

After data collection and image preprocessing, we pass the dataset into our model for training after data augmentation. In this work, we called up our generative neural network to produce data during training process and we have also defined our activation and optimization functions. In this study, we have incorporated the sigmoid activation function. Furthermore, we managed to compute the loss value by invoking the tf.nn.sigmoid_cross_entropy_with_logits() from TensorFlow library during training. With regard to discriminators, the anticipation is that real input needs to be close to 1; moreover, the outcome preceding from the generative model is expected to be 0. With regard to the generator network, the discriminator should produce a prediction of 1 for its generated images. The d_loss_real denotes the cross-entropy resulting from the discriminator's real data input and the expected result. Furthermore, 1. D_loss_fake represents the cross entropy arising from the difference between the originated data from the generator, the discriminator and the expected result. 0. D_loss connotes to the sum of d_loss_real and d_loss_fake. More so, G_loss refers to the cross-entropy arising from the difference between the results of generated data of the generator input the discriminator. Furthermore, the elected optimization algorithm in this situation is the Adam Optimizer,[29] which accommodates the non-convex optimization characteristics suitable for modern deep learning. With reference to that, there is no need to manually modify the learning rate and additional hyperparameters. The discriminator employs a cross-entropy loss function based on number of inputs that were accurately classified as real and number of inputs precisely categorized as generated during training.

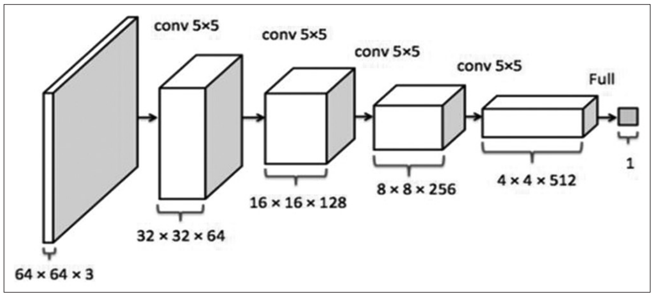

Figure 10: Discriminator

Results

Produced images

The adversarial network utilizes random noise as its input and outputs the ultimate prediction of the discriminator on the produced images. By adjusting the noise vector, we can get some profound knowledge of how the generator operates and discerns which noise vector results in our desired class. In this case, we discover by trying multiple noise vectors. The more we tried the output was getting only better at producing more abstract and distinct background colours with no white spaces. However, in some projects it was not certainly the case, for example, the sample produced from CIFAR-10[30] produced more white spaces which affected the quality of the generated images. The majority of GAN architectures use 100 as their input shapes, so initially, we had used the same but when we later amended it to 128 and it made some improvements to our results. We have utilized the trained generator to create authentic images based on the random sample noise depicted in Figure 11.

The following suggests different depictions of generations from the generator at different loop variations. In coming

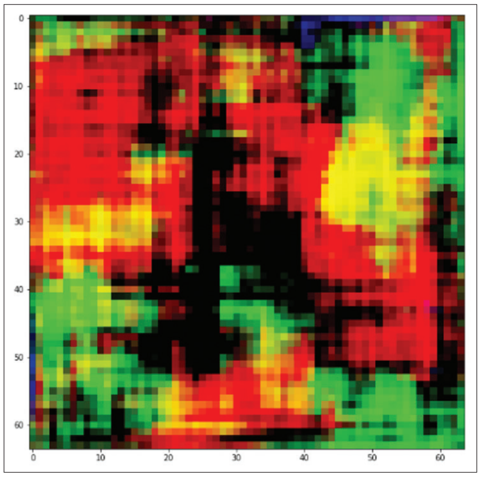

Figure 11: Random noise vector sample as input to the generator

up with the results we have incorporated a great deal of tuning to get it running. We tried numerous and different setups and observing the results while tweaking different components such as the hyperparameters, loss calculators, optimizers, learning rate, and activators. It was the best way to enhance our understanding of the algorithm parameters before we got the desirable results. The examples of images produced after several iterations of training are exhibited in Figure 12. It is significant to look at plots of these generated images at each point to enable us to see the progression in our generated images. At an early stage of the training, the generative model images are of low quality with a great deal of noise, and we can see that the generator has learned to generate a few delicate features in brown texture in Figure 13. The model begins to generate very conceivable lesions with repeated noise textures after 200 epochs. The generated images after 400 epochs are not significantly different, but we can visibly start to detect lesion edges. We observe that in each step there is a smooth transition in resolution, the image quality is enhanced in Figure 14, permitting the model to fill

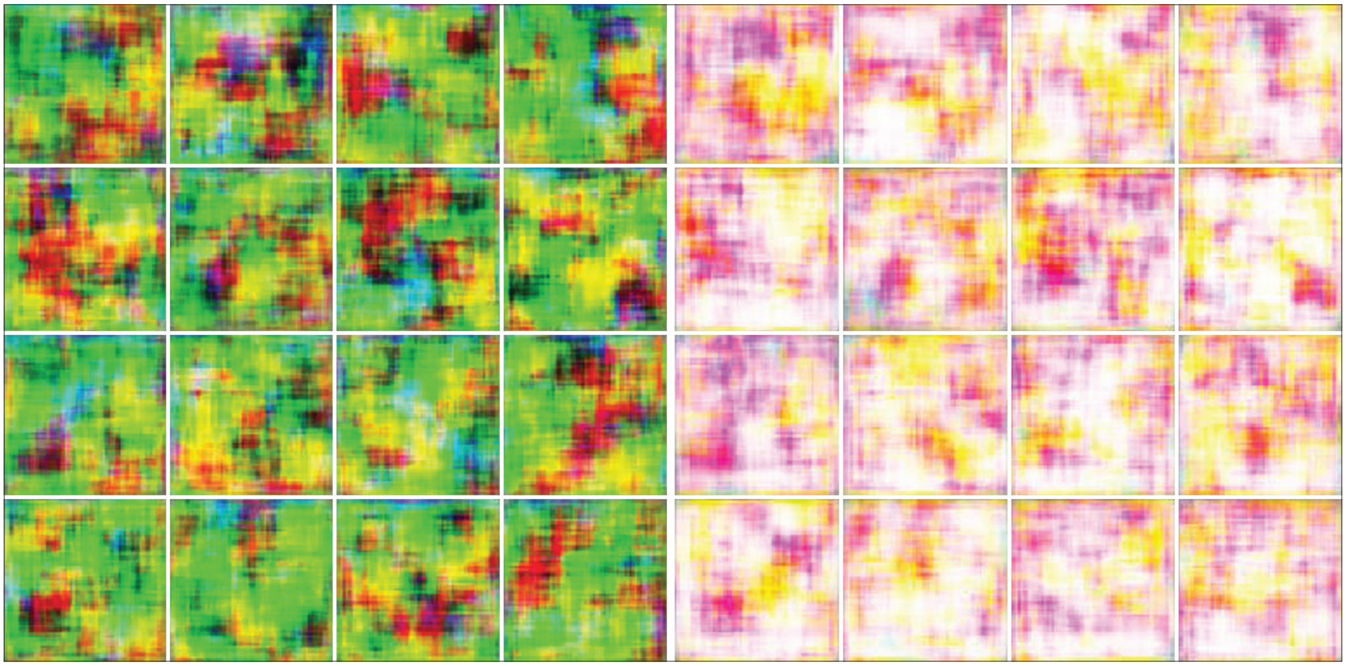

Figure 12: Generated skin cancer images after 400 epochs of training

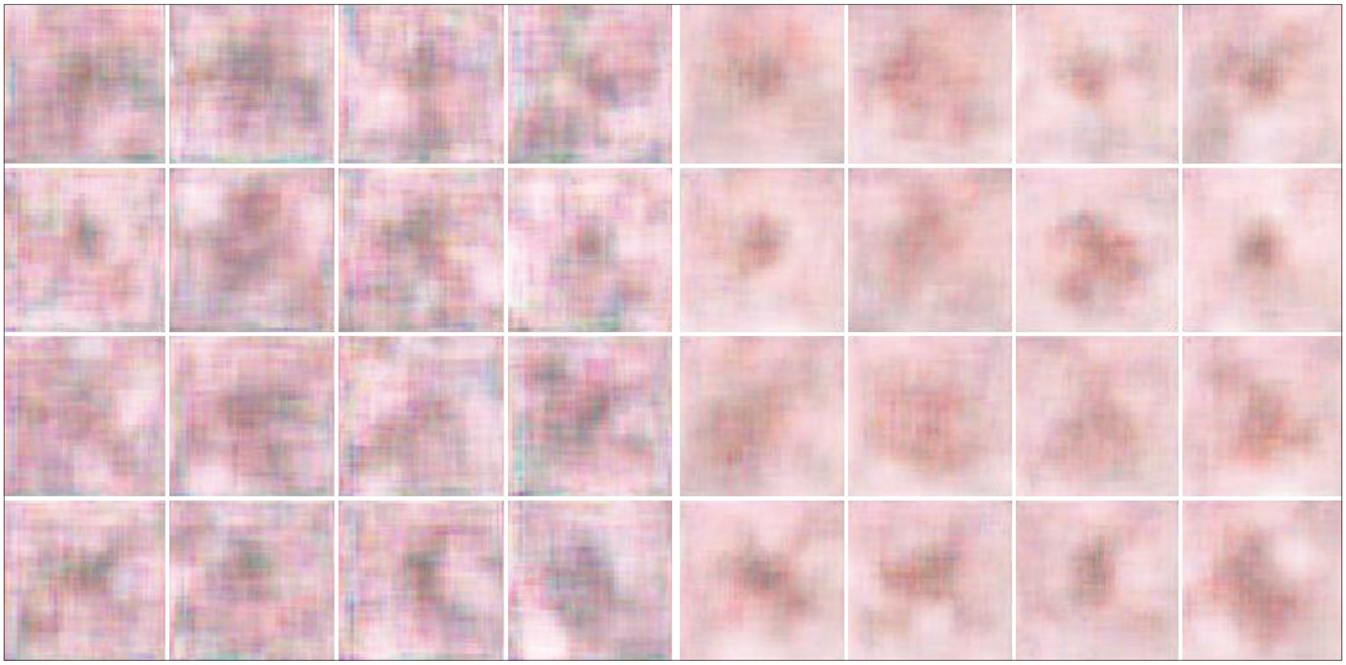

Figure 13: Generated skin cancer images after 1000 epochs of training

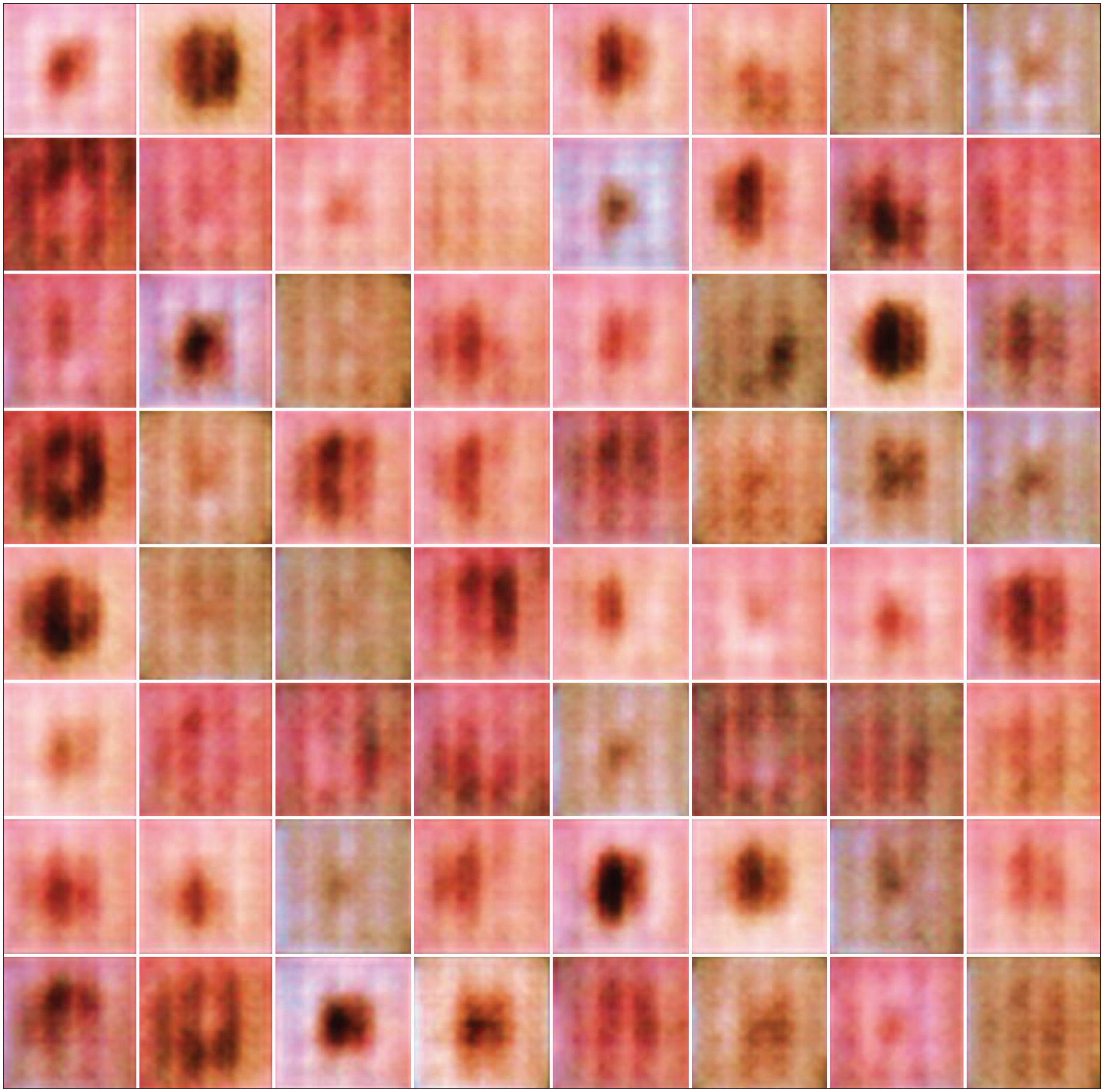

Figure 14: Generated skin cancer images after 3000 epochs of training

in more structure and detail to depict the lesion of our desired class.

Our proposed model was knowledgeable to produce images that are closely similar to synthetic pigmented lesions. The samples produced after several iterations of training are exhibited in Figure 15. Moreover, it is interesting to note the model was able to give a fairly reasonable deconvolution performance that is even better than those of models trained with labels such as in MNIST.[19] Figures 12 and 13 show example images produced by the generator over the course of training iterations, from earliest to latest. In Figures 12 and 13 we can view that our generator started producing little more than random noise. Throughout the training iterations, it improved more and more at emulating the features of our training data. The generator improved a little, each time our Discriminator rejected a false-generated image. We demonstrated that an unsupervised DCGAN trained on a large image dataset can also learn a considerable number of features that are interesting. We have employed a bilateral filter to enhance our image data and it retained more details to improve the quality of rendered images to achievable classification accuracies. However, after examining closely into the results of the resolution of our images, we observe

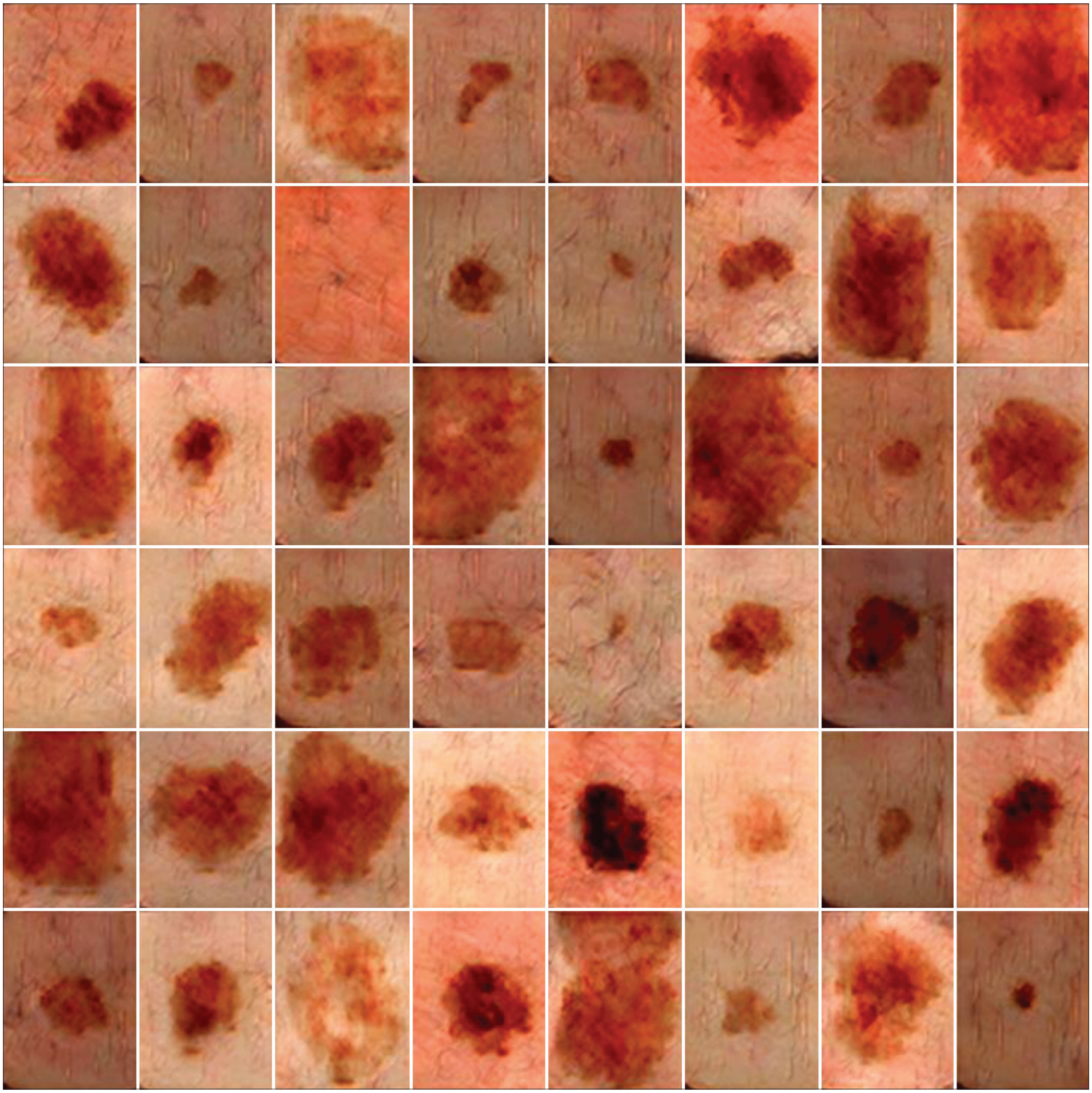

Figure 15: Generated skin cancer images after 3000 epochs of training

that the completed images are not highly accurate compared with the original images. This is maybe due to the fact that the initial images have very few pixels in the initial phase. This suggests that if we were to train our model on higher resolution images it would have achieved a better performance. Data augmentation contributes to classification optimizations mostly; however, not in a coherent manner. Furthermore, if a model can perform reliably on augmented data, it can be a sign of efficiency, if we are to compare with the training images of MNIST, for example, digit '9' after augmentation may yield to a different classification result.

Loss in training

Below Figure 16 is the plot of the training losses for the generator G and discriminator D put on record after each iteration. Preferably, the generator should receive enormous random noise as its input sooner in the training because it needs to learn how to generate authentic data. The discriminator on the contrary does not always acquire large samples early on, because it may easily distinguish real and fake images. In addition, during training, the generator and discriminator also may face the risk of overpowering each other. It has been observed in Figure 12 that if the generator becomes too accurate, it will tenaciously harness shortcomings in the discriminator which then leads to undesirable results, whereas if the discriminator becomes too accurate, it will return values that are close to 0 or 1.

To train correctly, we had to make sure that the generator and discriminator are on a similar level throughout the training process. During the experiments, it can be noticed from the Figure 12, that while the discriminator continues to have a lower loss, the generator managed to overwhelm the discriminator and produced a fair result. Our generator was proficient enough to trick the discriminator thus proving that the discriminator was not able to extract the finer detail features in the skin lesion image data during the training process. This can entail the need to increase our dataset to allow our discriminator to learn more during the training process.

Accuracy

Accuracy is defined as the ratio of correctly classified samples out of all samples:

$$\text{Accuracy} = \frac{\text{Numberof CorrectlyClassifiedSamples}}{\text{NumberofAllSamples}} \quad (3)$$

We have used Classification Accuracy method to find the accuracy of our model. Classification accuracy is the ratio of number of correct predictions over the total number of all examples. In order to check the correct prediction, we incorporated the confusion matrix which adds the predicted results diagonally which will be the number of correct predictions and then divide them by total number of predictions as shown in Figure 17. In this binary classification, our accuracy value was calculated with respect to negatives and positives given in Table 1, as FP = false positives, FN = false negatives, TP = true positives and TN = true negatives.

Confusion matrix

To get a better understanding on confusion matrix shown in Figure 17, we can say the matrix gives us information about how our model has managed to classify malignant or benign lesion correctly. In this case if the image which has a malignant lesion is correctly classified as malignant it is deemed to be true positive and an image harboring a malignant lesion but diagnosed with benign lesion is deemed to be a false negative. Similarly, the image harboring benign lesion and is correctly diagnosed as benign is referred to as false positive and an image which has a benign lesion but diagnosed as malignant is qualified as false negative. Confusion matrix was particularly useful for measuring our AUC-ROC curve, specificity, accuracy, recall, and precision. The Discriminator's confusion matrix is expressed in the form of a tabular representation of all the possible outcomes, as given in Figure 17.

Receiver operating characteristic-area under the curve

Moreover, we have utilized also ROC curves to assess the performance of our classifier over its complete operating range. A ROC curve shown in Figure 18 is a plot which we have used to summarize and to understand the performance of our binary classification model particularly on the positive class.

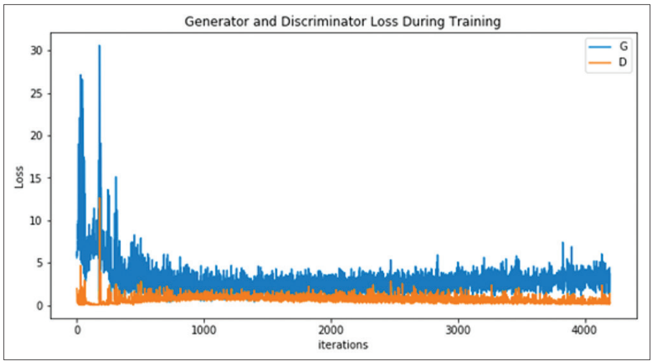

Figure 16: Training loss

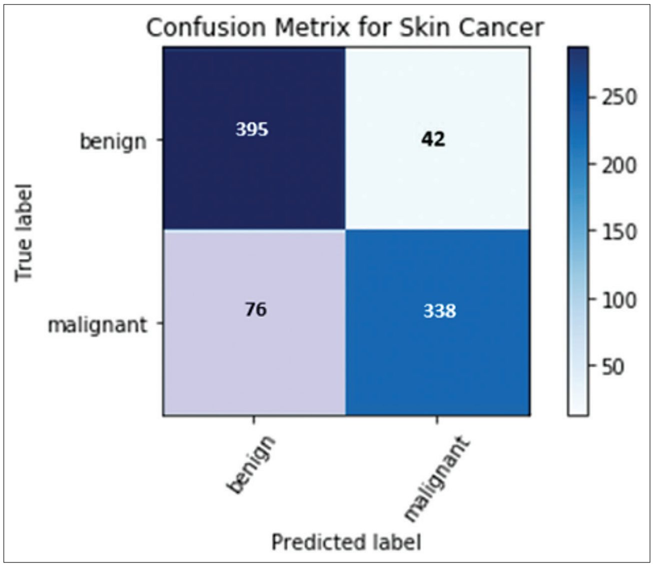

Figure 17: Confusion matrix

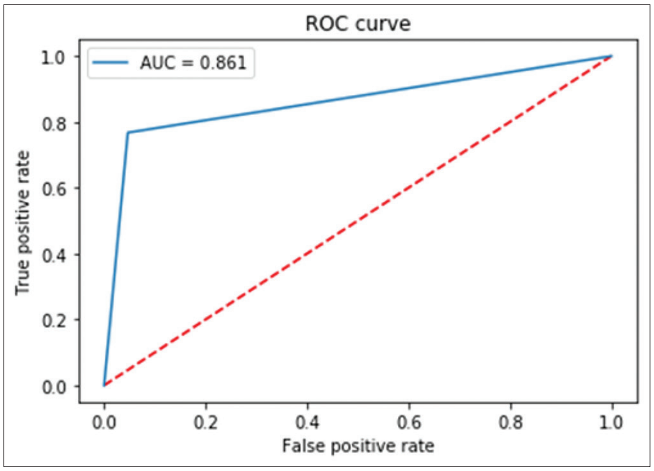

Figure 18: Receiver operating characteristic-area under the curve

Table 1: Accuracy variables

Predicted values	Actual values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Table 2: Effect of batch size on results

Batch size	Accuracy	Precision	Recall	F1 score	ROC-AUC
100	0.903	0.91	0.84	0.87	0.844
128	0.935	0.94	0.77	0.91	0.861

ROC-AUC – Receiver operating characteristic-area under the curve

Table 3: Effect of learning rate on results

Learning rate	Accuracy	Precision	Recall	F1 score	ROC-AUC
0.001	0.903	0.94	0.84	0.87	0.861
0.0002	0.915	0.88	0.72	0.79	0.813

ROC-AUC – Receiver operating characteristic-area under the curve

Table 4: Comparison of Adam versus stochastic gradient descent as optimizer

Optimizer	Accuracy	Precision	Recall	F1 score	ROC-AUC
Adam	0.935	0.94	0.84	0.87	0.861
SGD	0.903	0.92	0.71	0.80	0.830

SGD – Stochastic gradient descent; ROC-AUC – Receiver operating characteristic-area under the curve

Table 5: Comparison of related work

	Accuracy (%)	ROC-AUC
Haenssle *et al*.[31]	-	0.86
XG *et al*.[2]	81.33	-
JB *et al*.[29]		0.9150
This work	93.5	0.861

ROC-AUC – Receiver operating characteristic-area under the curve

Table 6: Fréchet inception distance comparison of related work

Model	DCGAN (ours)	GAN[21]	LAPGAN[16]
FID	44.41±1.28	50.93±1.18	41.88±1.08

FID – Fréchet inception distance; GAN – Generative Adversarial Network; DCGAN – Deep Convolution GAN; LAPGAN – Laplacian GAN

In this plot, y-axis implies the true positive rate (TPR) and the x-axis indicates the true false-positive rate (FPR). The dotted red line in our plot indicates the ROC curve of a random classifier. In Figure 14, our ROC curve depicts the trade-off between TPR, which is the sensitivity and (1– FPR) representing the specificity. In addition, usually a good classifier depicts a curve that is closer to the top-left corner which entails a better performance. More so, a less accurate test is depicted by a curve which is close to the 45° diagonal of the ROC space. As depicted by Figure 14, the classifier gave us a fair classification considering that the distance between the top left corner and our curve is small and also that our ROC curve deviated from the diagonal. Based on positives and negatives in our confusion matrix, our binary classification accuracy can be calculated as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{395+338}{395+338+42+76} = 0.861 \qquad (4)$$

$$Precision = \frac{TP}{TP+FP} = \frac{395}{395+42} = \frac{395}{437} = 0.903 \qquad (5)$$

$$Recall = \frac{TP}{TP+FN} = \frac{395}{395+76} = \frac{395}{471} = 0.839 \qquad (6)$$

$$F-score = 2\times\frac{Recall\times Precison}{\mathrm{Recall}+\mathrm{Precision}} = 2\times\frac{0.839\times 0.903}{0.839+0.903} = 2\times\frac{0.758}{1.742} = 0.87 \qquad (7)$$

The threshold utilized for classification varies between 0 and 1, and the sensitivity and specificity are determined for each selected threshold as depicted in Table 1. The accuracy calculated gave us 0.861, which 86% correct prediction out of total samples. This means that our lesion classifier did fairly good in distinguishing malignant lesions from benign lesions. Furthermore, our model produced a precision of 0.9 as stated in Table 2, which entails that each time it predicts that the image is comprised of malignant lesion, it is 90% correct most of the time. In addition, our model produced a recall of 0.83 which also entails that it has correctly identified 83% of all malignant lesions. To fully evaluate the effectiveness of our model, it is essential to scrutinize precision and recall at the same time but regrettably, there is always tension between the two, which arises from improving one of them. That is, improving recall typically diminishes precision and conversely. F-score is usually used to measure accuracy of our test basing on the precision and recall tests. Below are some changes made during training and their corresponding accuracy precision and recall values.

In Table 2 we can see that a batch size of 128 is giving us better results when compared to that of 100.Here we can see that our accuracy is fair enough considering the number of images on which our model is trained on. Bi *et al*. produced much better ROC-AUC. Besides that our work archives good results on skin cancer classification.

Batch size is an essential hyperparameter to tune in modern deep learning paradigms. In most cases smaller batch sizes tend to permit the model to learn the pattern in the data without having to train on a big dataset. However, in this study we started with a small batch size, and we started reaping some benefits steadily as we increased our batch size as depicted in Table 2.

From our results we can tell that the batch size has a significant impact on the corresponding accuracy precision and our performance in general. More so our learning rate may determine how we are going to converge. To travel slowly on the downward slope, we used a low learning rate just as depicted in Table 3, we made the value low to try to converge to an optimal point.

We can also conclude that the learning rate is correlated to batch size because when we increased the batch size and lowered our learning rate the performance was enhanced. In Table 4 ,Adam Optimiser seem to be working better than the Stochastic Gradient Descent (SGD) because the SGM goes through a lot of iterations before it can reach the optimal point and its randomness during its descent affects its performance when converging. In Table 5. Here we can see that our accuracy is fair enough considering the number of images on which our model is trained on. Bi et al produced much better ROC-AUC.In addition, we measured our FID score based on same number of real samples and fake and our work achieved good results on skin cancer classification as depicted in Table 6. Our model achieved lower FID score which depicts a better performance. FID score is mostly used to measure the diversity and quality of images.

Fréchet inception distance

In 2017 Heusel[33] al introduced Fréchet inception distance (FID), which was used to estimate realism by measuring the distance between the distribution of the generated images and the true distribution. FID is used to measure the quality and diversity of images and it needs a more decent sample size to produce good results. Too few samples will cause an over-estimating of the actual FID and consequently the estimates will depict a large variance.

It must be noted with concern, that it was difficult to determine the diversity of our output images based several training images we implemented.[34] However, we managed to conduct a quantitative experiment making use of renowned FID. The FID score is commonly used to measure the diversity and quality of images. The Equation to calculate FID score is calculated as below:

$$FID = \mu_x - \mu_y \frac{2}{2} + Tr\left(\sum_x + \sum_y - 2(\sum_x \sum_y)^{\frac{1}{2}} \right) \tag{8}$$

Where x and y represents the sets of images. In this experiment, x and y comprises of real and fake images, respectively. In this representation, mean (μ) and variance (Σ) is used to portray the visual quality and diversity of our images. A low FID score denotes that these two sets of images have a similar probability distribution. Therefore, a lower FID score represents a better performance which can be used to compare with other comparable models. We measured our FID score based on same number of real samples and fake and the result is summarized in Table 1.

We can see here that our model produces a fair result as compared to other models given in the above table. It could have performed more better given more computational power.[35-38]

Difficulties and shortcomings

The constraints with this study was mainly to build a proper Deep Convolutional Generative Adversarial Network (DCGAN) and make it generate high quality images that are much like the synthetic real samples. Generally, deep learning models have great deal of model parameters as well as a plenty of hyper-parameters to be adjusted and we had to be very careful in tuning these hyper-parameters. This made the fine tuning of our model very time consuming due to the many tests that needed to be performed. Speaking of time, the time required for each training session was the foremost limitation to this study.[39-41] One training session for skin lesion dataset could require 1 h and 30 min, the time could rise up to 3 h per session. The extremely time-consuming training sessions took up most of the time dedicated for this study. This further complicated the fine-tuning of the DCGAN on the dataset. Another challenge was constraints in GPU power, we were unable to generate a perfect sample based on the previous original image data. We would have wished to have train our model on a more robust RTX 2080 Ti GPU because it offers an excellent performance in deep learning.[42,43]

Conclusions and Future Work

Conclusions

In this endeavor, we focused on investigating the ability of deep convolutional neural networks to discriminate between malignant and benign cancer and at the same time trying to overcome the limitations of GAN models. The main purpose of this work was to improve the overall accuracy level by using the refined set of skin cancer images obtained after applying some common image preprocessing algorithms such as gamma correction and bilateral filter. The experiments were conducted using the Keras python framework and the proposed method was applied to the Kaggle Skin Cancer Dataset which had the same number of samples for both classes.

We demonstrated that a typical deep neural networks-GAN method can attain competitive classification performance and produces some better diagnostic accuracy that can outperform contemporary methods, expert physicians and clinicians in skin lesion classification other. In this work, the experimental results are very promising, especially after some noise removal algorithms were incorporated. We have seen some substantial improvement in training after applying image enhancement and pre-processing operation. Thus, we have achieved an overall test accuracy of 93.5% after fine-tuning most parameters of our network.

Future work

In the future work, firstly, we consider that models carry out enormous amounts of computations during the training process. DCGAN contains two models which means that even more computations are performed so large computational power is required to reduce the instability issues related to GAN models. The principal contribution of the suggested course of work is hypothetical, and it can be implemented to other kinds of images as well. Consequently, it clarifies the generalization of the model for different more cases than the original model. Probably using a larger training set or training the model a little longer may attain the desired results. While many different adaptions, tests and experiments have been performed, there is still room for changes in the future. Especially the fine-tuning of the DCGAN on medical photos or other similar but less noisy datasets is interesting since there were no time left for the researcher to continue. Also finding other ML tasks than classification where the accuracy is very dependent to the number of examples available in order to investigate how synthetic examples generated by a GAN affects the accuracy. The author suggests that before a dataset or a task is selected, one should examine if the accuracy is highly dependent on the number of training examples. Another proposal is to investigate whether a DCGAN can be trained to generate noisy samples in order to prevent overfitting.

Furthermore, additional investigations into the internal structure of the network to manipulate the generator representation might be the next steps in this study. In addition to that, widening this work would be very interesting by exploring images in their grey level and binary state to extract the desired lesion or separating them from the background of the image upon inspection.

Irregular shapes of skin lesions, different types of colors on each skin, and determining the region of interest on each dermoscopy image are just a few challenges in skin cancer detection. Detecting minute changes on the skin requires expertise in this field. However, the human eye may not always catch these tiny changes. Helping doctors with the computer vision and deep learning techniques can save many lives. With this motivation, we studied skin cancer malignancy detection to classify skin lesions and identify malignant cases. Pretraining settings and posttraining measurements of all experiments showed that the skin cancer malignancy detection is a difficult task and generalizing a model for all cases requires some image preprocessing techniques to apply before feeding into any deep learning algorithm. We did many experiments and tried various techniques to solve the complexity of skin lesions classes. This result is a good indicator for the potential of such a technology to reduce false-negative and false-positive predictions and eventually help physicians increase their diagnostic prediction power.

Financial support and sponsorship

None.

Conflicts of interest

There are no conflicts of interest.

References

1. Hay RJ, Augustin M, Griffiths C, Sterry W, Board of the International League of Dermatological Societies and the Grand Challenges Consultation groups. The global challenge for skin health. Br J Dermatol 2015;172:1469-72.
2. Lopez AR, Giro-i-Nieto X, Burdick J, Marques O. Skin lesion classification from dermoscopic images using deep learning techniques. 13th IASTED Int Conf Biomed Eng (BioMed*)* 2017. p. 49-54.
3. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, *et al*. Generative adversarial nets. In: NIPS’14: NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014;2:2672–80.
4. Dong HW, Yang YH. Training Generative Adversarial Networks with Binary Neurons by End-to-end Backpropagation. Taipei, Taiwan: Research Center for IT Innovation, Academia Sinica; 2018
5. Yap J, Yolland W, Tschandl P. Multimodal skin lesion classification using deep learning. Exp Dermatol 2018;27:1261-7.
6. Aldwgeri A, Abubacker NF. Ensemble of Deep Convolutional Neural Network for Skin Lesion Classification in Dermoscopy Images . International Visual Informatics Conference, IVIC 2019: Advances in Visual Informatics. p. 214-26.
7. Khan MQ, Hussain A, Rehman Su, Khan U, Maqsood M, Mehmood K, *et al*. Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer. July 2019IEEE Access PP(99):1-1, DOI: 10.1109/ACCESS.2019.2926837.
8. Pham TC, Luong CM, Visani M, Hoang VD. Deep CNN and data augmentation for skin lesion classification. Intell Inf Database Syst Lect Notes Comput Sci 2018;10752:573-82.
9. Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, *et al*. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 2016;35:1285-98.
10. Karabulut E, Ibrikci T. Texture analysis of melanoma images for computer-aided diagnosis. Int Conf Intell Comput Comput Sci Inform Sys (ICCSIS 16) 2016;2:26-9.
11. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, *et al.* Dermatologist-level classification of skin cancer with deep neural networks. Nature 2017;542:115-8.
12. Yang X, Zeng Z, Yeo SY, Tan C, Tey HL, Yi Su. A novel multitask deep learning model for skin lesion segmentation and classification. arxive 2017;1:10-5.
13. Nasr-Esfahani E, Samavi S, Karimi N, Soroushmehr SM, Jafari MH, Ward K, *et al*. Melanoma detection by analysis of clinical images using convolutional neural network. Annu Int Conf IEEE Eng Med Biol Soc 2016;2016:1373-6.
14. Hosny KM, Kassem MA, Foaud MM. Classification of skin lesions using transfer learning and augmentation with Alex-net. PLoS One 2019;14:e0217293.
15. Radford A, Metz L, Chintala S. Unsupervised Representation Learning,” Under Review as a Conference Paper at ICLR 2016 Indico Research; 7 January, 2016:3-15.
16. Denton E, Chintala S, Szlam A, Fergus R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. New York: Facebook AI Research; 2015. p. 12-20.
17. Ahirwar K. Build next-generation generative models using TensorFlow. In: Generative Adversarial Projects. Packt Publishing Ltd; 2019. p. 89-95.
18. Tsang SH. “Review: DCGAN –Deep Convolutional Generative Adversarial Network (GAN).” Available from: https://medium.com/@sh.tsang/review-dcgan-deep-convolutional-generative-adversarial-network-gan-ec390cded63c. [Last accessed on 2020 Apr 25].
19. Nishad G. “MNIST-GAN: Detailed Step by Step Explanation & Implementation In Code,” INTEL AI ACADEMY. Available from: https://medium.com/intel-student-ambassadors/mnist-gan-detailed-step-by-step-explanation-implementation-in-code-ecc93b22dc60. [Last accessed on 2018 Aug 01].
20. Chen F, Chen N, Mao H, Hu H. Assessing four neural networks on handwritten digit recognition dataset (MNIST). Chuangxinban

J Comput 2018.

21. Qin Z, Liu Z, Zhu P, Xue Y. A GAN-based image synthesis method for skin lesion classification. Comput Methods Programs Biomed 2020;195:105568.
22. Cronin J, Finni T, Seynnes O. Using deep learning to generate synthetic B-mode musculoskeletal ultrasound images. Computer Methods and Programs in Biomedicine 2020;196:DOI:10.1016/j.cmpb.2020.105583.
23. Ghassemi N, Shoeibi A, Rouhani M. Deep neural network with generative adversarial networkspre-training for brain tumor classification based on MR images. Biomed Signal Process Control 2020; 57: DOI: 10.1016/j.bspc.2019.101678.
24. Github, "Github," Github; 31 Jan 2018. Available from: https://github.com/4thgen/DCGAN-CIFAR10. [Last accessed on 2020 May 26].
25. Kaggle, "Kaggle," Kaggle; 20 Sep 2016. Available from: https://www.kaggle.com/c/cifar-10. [Last accessed on 2020 May 10].
26. Fanconi C, "Kaggle," Kaggle; 28 november 2018. Available from: https://www.kaggle.com/tatsukiishikawa/skin-cancer-classification. [Last accessed on 2020 May 14].
27. Gedraite ES, Hadad M. Investigation on the effect of a gaussian blur in image filtering and segmentation. Proceedings ELMAR-2011.
28. Adlam B, Weill C, Kapoor A. Investigating under and overfitting in wasserstein generative adversarial networks. arxive.org, 2019;2:12-22.
29. JB, Kingma DP. Adam: A Method for Stochastic Optimization. Published As a Conference Paper at ICLR 2015; 2017.
30. Desai U. "Training a Conditional DC-GAN on CIFAR-10," Medium; 08 June 2018. Available from: https://medium.com/@utk.is.here/training-a-conditional-dc-gan-on-cifar-10-fce88395d610. [Last accessed on 2020 Apr 20].
31. Haenssle HA, Fink C, Schneiderbauer R, Toberer F, Buhl T, Blum A, *et al*. Man against machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. Ann Oncol 2018;29:1836-42.
32. Yu L, Chen H, Dou Q, Qin J, Heng PA. Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks. IEEE Transactions on Medical Imaging 2017;36:DOI:10.1109/TMI.2016.2642839.
33. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems 30. NeurIPS Proceedings Search 2017:2:9-15.
34. Ozkan IA, Koklu M. Skin Lesion Classification using Machine Learning Algorithms. IJISAE 2017;5:285-9.
35. Moffy Crispin Vas D. Classification of benign and malignant lung nodules using image processing techniques. Int Res J Eng Technol 2017;4:DOI:10.1088/1361-6560/ab2544.
36. Lei B, Xia Z, Jiang F, Jiang X, Ge Z, Xu Y, *et al*. Skin lesion segmentation via generative adversarial networks with dual discriminators. Medical Image Analysis 2020;64:10.1016/j.media.2020.101716.
37. Fang W, Zhang F, Sheng VS, Ding Y. A method for improving CNN-based image recognition using DCGAN. Computers Materials and Continua 2018;57:167-78.
38. Liu S, Yu M, Li M, Xu Q. The research of virtual face based on Deep Convolutional Generative Adversarial Networks using Tensorflow. Physica A: Statistical Mechanics and its Applications 2019;521:667-80.
39. Haputhanthri U. "Medium," The AI Team; 17 MAY, 2020. Available from: https://mc.ai/introduction-to-deep-convolutional-generative-adversarial-networks-using-pytorch/. [Last accessed on 2020 May 25].
40. Yadav V, Kaushik VD. A study on automatic early detection of skin cancer. Int J Advanced Intelligence Paradigms 2019;12, Nos. 3/4, 2019 .
41. World Cancer Research Fund International, "WCRF.Org;" 2018. Available from: http//www.wcrf.org. [Last accessed on 2020 May 03].
42. Email MP, Simson W, Guha A, Rüdiger R, Christian G, Navab WN. Manifold Exploring Data Augmentation with Geometric Transformations for Increased Performance and Robustness, International Conference on Information Processing in Medical Imaging IPMI 2019: Information Processing in Medical Imaging p. 517-29.
43. Hwang U, Choi S, Lee HB, Yoon S. Adversarial training for disease prediction from electronic health records with missing data. arXiv preprint arXiv:1711.04126