Arun Solanki
Mohd Naved *Editors*

GANs for Data Augmentation in Healthcare

Springer

GANs for Data Augmentation in Healthcare

Arun Solanki • Mohd Naved
Editors

GANs for Data Augmentation in Healthcare

Springer

Editors
Arun Solanki
Gautam Buddha University
Greater Noida, India

Mohd Naved
SOIL School of Business Design
Greater Noida, India

ISBN 978-3-031-43204-0 ISBN 978-3-031-43205-7 (eBook)
https://doi.org/10.1007/978-3-031-43205-7

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Contents

Role of Machine Learning in Detection and Classification of Leukemia: A Comparative Analysis

Ruchi Garg, Harsh Garg, Harshita Patel, Gayathri Ananthakrishnan, and Suvarna Sharma

1 Introduction

The discipline of bio informatics involves using computation to draw conclusions from a variety of biological data [1]. Through the application of algorithms, it ensures the collection, archival, retrieval, manipulation, and modelling of data for analysis, prediction, or visualization. The size and quantity of biological datasets that are currently available have greatly risen in recent years, which has prompted bio informatics researchers to apply a variety of machine learning and data mining algorithms [2]. Deep learning and other machine learning approaches are currently being used for autonomous feature learning with datasets for various biological

Ruchi Garg, Harsh Garg, Harshita Patel, Gayathri Ananthakrishnan, and Suvarna Sharma have contributed equally to this chapter

R. Garg
Information Technology, Birla Institute of Management Technology, Greater Noida, Uttar Pradesh, India
e-mail: ruchi.garg@bimtech.ac.in

H. Garg
Department of Electrical Engineering, Delhi Technological University, New Delhi, India

H. Patel (✉) · G. Ananthakrishnan
School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamil Nadu, India
e-mail: harshita.patel@vit.ac.in; gayathri.a@vit.ac.in

S. Sharma
Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India
e-mail: suvarna.sharma@chitkara.edu.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_1

research projects [3–5]. The work presented in this chapter is on the diagnosis of leukemia, a lethal disease of white blood cells (WBC) that affects the blood and bone marrow in humans [6] using various machine learning algorithms. There are two different types of leukemia: acute and chronic [7]. The type of leukemia is determined by several genetic variations and gene expressions with their gene value related with the white blood cell.

Uncontrolled accumulation of aberrant white blood cells is a defining feature of the cancerous illness leukemia. Leukemia is a fatal kind of cancer. Hematopoiesis, which occurs in the bone marrow, which fills the bone's interior cavity, is the process by which all blood cells form. Acute leukemia causes the patient to rapidly deteriorate, whereas chronic leukemia progresses slowly and might be lymphocytic or myelogenous. The World Health Organization (WHO) proposal and the French-American-British (FAB) classification are the two classifications now in use to categorize leukemia [8]. Acute myeloid leukemia (AML) [9, 10], which contains seven varieties (M-1-M-7), was identified by blast cells seen in a peripheral blood smear. This method is unsuitable for studying a large number of cells since it is boring, time-consuming, and laborious. However, several mathematical techniques and technologies have been created to differentiate between blood cells, which is crucial for detecting leukemia [11, 12]. These researchers [13] hypothesized a subjective mapping standard discovery model, and they concluded that learning from earlier mastery cannot be taken into account to better dictate the parameters because the ultimate objective is to improve meeting and shorten the learning period. We have investigated a number of well-known machine learning techniques, such as decision trees, support vector machines, and k-nearest neighbors. In order to determine which machine learning (ML) method will deliver the highest level of accuracy for the used dataset, the objective of this study is to evaluate several ML algorithms. In order to determine which machine learning algorithm can provide the maximum accuracy for a dataset to determine if a patient has acute leukemia or chronic leukemia, detailed comparison of several machine learning algorithms has been done. The methods used in this research include k-Means, Naive Bayes, support vector machine (SVM), logistic regression, and XG-Boost.

2 Methodology

The Jupyter notebook software on Windows OS is used to implement the various algorithms on the dataset. The dataset is uploaded after importing the libraries. The dataset was trained. Following that, many machine learning methods are applied to the dataset, including k-Means, SVM, logistic regression, Naive Bayes, and XG-Boost. A measurement is made of each algorithm's accuracy. As expected from the theory, each algorithm's accuracy value came out to be almost different. Besides testing the accuracy, an additional work of creating the confusion matrix has also been done in this chapter. Confusion matrix is created for each of the methods. The confusion matrix provides a visual representation of the nature of predictions of the

work done in this paper. Additionally, it will aid in identifying the algorithm that will prove to be the most accurate. This approach may be used to find the algorithm that will work best to identify the type of leukemia the patient has, allowing for the earliest possible treatment. Only the gene expression value may simplify our effort without using the conventional way of gathering and analyzing blood samples since we can apply several machine learning algorithms to find the optimal algorithm that can produce the result with the greatest accuracy. The dataset which is used and applied on the algorithms has been taken from the link as mentioned: https://github.com/titichhm/AI-Project-Dataset. The various tools which have been used to conduct the work are Numpy, Pandas, Matplotlib, Seaborn, mpl toolkits, Keras, scikit. learn, Tensor flow, and XG-Boost.

3 Literature Review

For the detailed study, a number of good research papers have been studied. The application of ML algorithms for the identification, classification, and diagnosis of leukemia illness has undergone a thorough, comprehensive examination. Elaborated analysis of the work is presented in this section. On the basis of the kind of algorithm employed, the literature review offered in this section is categorized into three subsections, as shown in Fig. 1.

According to the kind of algorithm employed, the literature review offered in this part is separated into two subsections.

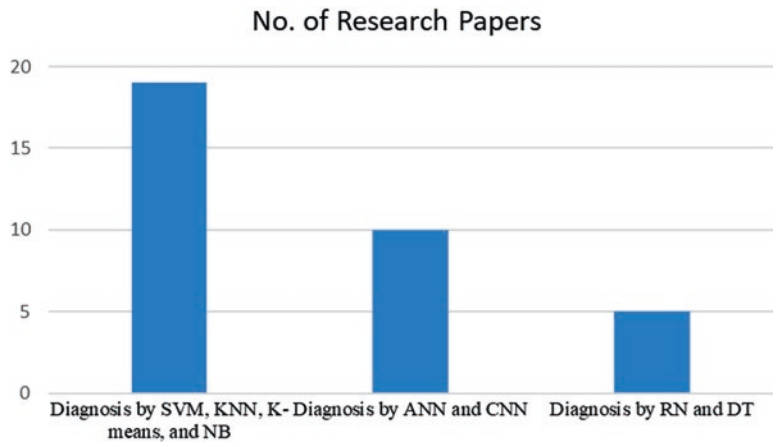


Fig. 1 Number of research papers reviewed

3.1 Diagnosis by Using SVM, KNN, K-Means, and Naive Bayes

Tanima Thakur et al. [14] show the use of gene expression data in predicting cancer as an important development in the field of cancer diagnosis and treatment. Despite the variety of approaches available, it is crucial to select one that is appropriate for the dataset being used and to weigh the benefits and drawbacks of each method. For these techniques to be more accurate and available to patients and healthcare professionals on a larger scale, additional study is required.

In the work presented by Muhammad Hammad Waseem et al. [15], there are 72 individuals in the leukemia dataset, of whom there are 47 cases of acute lymphoblastic leukemia (ALL) and 25 cases of acute myeloid leukemia (AML). Waseem et al. [16] employed 100 blood pictures (acute lymphoblastic leukemia, acute myeloid leukemia, chronic lymphocytic leukemia, and chronic myeloid leukemia). SVM, short for support vector machine, is employed. While identifying leukemia from blood pictures, SVM has produced results with an overall accuracy of 79.38%.

A collection of 130 ALL contaminated pictures was used by Jyoti Rawat et al. [17]. A total of 65 of these photos have been utilized for training. The remaining images were employed to test the suggested work. Gray level cooccurrence matrices (GLCM) and an automatic support vector machine (SVM) binary classifier have been applied. For the cytoplasm and nucleus, respectively, texture-based feature categorization accuracy is 86.7 and 72.4%. However, classification accuracy of 56.1 and 72.4%, respectively, were found for shape-based features. Classification accuracy is attained with combined texture-shape feature of 89.8%.

In the research work presented by Ahmed S. Negm et al. [18], a dataset having 757 images is used. Leukemia cells are located utilizing panel selection, segmentation using K-means clustering, feature mining, and picture refining techniques. According to test findings, the accuracy was 99.517%, the sensitivity was 99.348%, and the specificity was 99.529%.

In the work of Ahmed M. Abdeldaim et al. [19], dataset consisting of 260 cell images is used. Out of these, 130 images were of normal subjects whereas the rest of the 130 were of those affected by acute lymphoblastic leukemia (ALL). The classifiers of k-nearest neighbors (KNN) and SVM are used by the researchers. The accuracy shown by KNN is 93.2% whereas the classification accuracy given by SVM came out as 87.4%. Hence, KNN came out better as compared to SVM in this case. A dataset of images having a total of 2555 is used by Jaroonrut Prinyakupt et al. [20]. Out of these, 601 images comprised of white blood cells. 2477 cropped white blood cell pictures are included in the collection. Using Naive Bayes classifiers, the performance was compared. The total correction rate for Naive Bayes models during the classification phase is around 94%.

Oscar Picchi Netto et al. [21] have used a dataset of 72 samples. The remaining 34 samples are utilized for testing, while the remaining 38 samples are used to train the dataset. The techniques used by the researches are decision trees using nominal values and SVM. The accuracy given by decision trees is 94.2 whereas 86.4% is

achieved for SVM. The sample size of 126 consists of acute lymphoblastic leukemia and its subtypes is used by Amjad Rehman et al. [22]. Naive Bayesian classifiers, KNN, and SVM are used. SVM accuracy is 97.78%, KNN accuracy is 82.5%, and Naive Bayesian accuracy is 92.3%. The dataset used by Enrique J.deAndr'es-Galiana et al. [23] was of the year ranging from 1997 and 2007. This clinical data was gathered from Hospital Cabuen˜es (Asturias, Spain). This dataset has a sample size of 265 Caucasians. These patients were identified with CLL. The classifiers SVM and KNN are used for algorithmic prediction. KNN has a precision of 54.7% while SVM has a precision of 90.1%.

Juli'an Candia et al. [24] collected a dataset of 847. Human microRNAs were measured for each sample. Filtering is done on this dataset which resulted in a smaller set of 370 microRNAs. The samples having low or absent expression of microRNAs were removed. The researchers used support vector machine (SVM) with a linear kernel in their work. 99.5% is the prediction performance of the model. A sample size of 120 patients suffering from malignant neutrophils of chronic myelogenous leukemia (CML) is collected by Wanmao Ni et al. [25]. The method of support vector machine method is used by them. They recorded a high specificity $\leq$95.80% and sensitivity $\leq$95.30%.

The sample size total of 100 microscopic blood cell images were attained by Sachin Paswan et al. [26]. Image color threshold is used to preprocess the images. For segmentation, the threshold approach is employed. Hausdorff dimension, shape features, texture features, and GLCM were features taken into account throughout the computation. The classification process uses KNN and modified SVM. KNN reported a 61.11% accuracy rate. 83.33% accuracy was reported using SVM. The SVM method was enhanced, and an initialization phase to discover a 12-neighbor linked component was included.

Morteza Moradi Amin et al. [27] have made a dataset of 42 samples from two sources. These sources were Isfahan Al-Zahra and Omid hospital pathology lab. Dataset consists of 21 bone marrow slides and peripheral blood smears from 14 ALL patients and 7 healthy individuals. Segmenting the nucleus and preprocessing were done. Segmentation was performed using K-means. Features were extracted following creation and selection. In the first stage, traditional SVM was utilized. As there were six classes, a multi class SVM classifier was utilized in the second stage. Evaluation used the K-fold cross validation procedure with $k = 10$. Clarity, precision, and sensitivity for the binary SVM classifier were 98, 95, and 97%, respectively. These numbers are 84.3, 97.3, and 95.6% for the multi-class SVM classifier, respectively.

Sachin Kumar et al. [28] got the database from a hospital. Dr. RML Awadh Hospital in Lucknow provided samples for the suggested work. Preprocessing was done to improve the quality and reduce unwanted distortions. On a dataset of 60 pretested samples, the proposed technique was tested with KNN and Naive Bayes Classifier after feature extraction for a few chosen characteristics. 92.8% accuracy was attained. Both KNN and Naive Bayes classifiers obtained approximately the same sensitivity, while KNN had significantly higher specificity. Khaled et al. [29] in the work prepared a database of 4000 samples. It includes 200 samples of the

patients suffering from leukemia disease. Every sample has 18 attributes. Dataset was gathered from the European Gaza Hospital's CBC testing repository. To increase precision, data preparation was carried out. Three classifiers—SVM, DT, and KNN—were applied. To acquire the accuracy, they were applied using RapidMiner. The accuracy of DT was 77.30%, the highest of the three algorithms. In terms of outward qualities, it also acquired properties.

Rodellar et al. [30] analyzed a set of 9395 images for the image processing of morphological analysis of blood cells using various approaches for image processing and segmentation. The gray-level co-occurrence matrix and granulometry are the two primary methods for evaluating textures (GLCM). Support vector machines (SVM), decision trees, and neural networks are employed. To assess the effectiveness across various color photographs, multiple color models were evaluated in datasets. Minal et al. [31] collected two different kinds of datasets. Both the ALL-IDB1 and ALL-IDB2 have segmented WBCs to test the categorization of blast cells. Both of these databases may be used to evaluate the segmentation capabilities of algorithms, classification systems, and image preprocessing techniques. Blast cells have been distinguished from regular lymphocyte cells using the KNN classifier. Fuzzy C-Mean clustering and K-Mean clustering are utilized. Using a KNN classifier, leukemia detection with suggested characteristics was categorized, yielding an overall accuracy of 93%.

The UCI machine's cross-domain sonar data collection [8] learning repository is used to assess the suggested techniques. For training and testing, there are 140 and 68 occurrences, respectively. Bare-bones particle swarm optimization (BBPSO) approaches are introduced to identify the key distinctions between blast and healthy cells in order to efficiently categorize ALL. Lymphocytes are categorized using Gaussian Radial Basis Function (RBF), Support Vector Machine (SVM), 1-Nearest Neighbor (1NN), and the best feature subsets. Improved geometric mean performances of 94.94 and 96.25% are produced by the provided methods for the SDM-based clustering strategy.

Fatemeh Kazemi et al. [32] observed a total of 1500 data. Out of the dataset 750 were used for ALL and 750 were used for AML. In total 1500 were divided into 1200 train data and the remaining data as test data. Binary support vector machine (SVM) classifier, k-means clustering, and fuzzy C-means clustering applied to segregate the foreground and background are used to classify photos into malignant and noncancerous images. According to the findings, KNN performed well in categorizing both AML and ALL with high percentage accuracy up to 86%. Sachin Kumar et al. [28] collected a dataset of 6000 samples. The author applied the K-mean Clustering Algorithm to image processing. The approach makes use of basic enhancement, morphology, filtering, and segmentation techniques as well as the k-means clustering algorithm to identify regions of interest. The suggested approach was evaluated using Nearest Neighbor (KNN) and Naive Bayes Classifier, and it demonstrated an accuracy of 92.8%.

The dataset that was made accessible [33] was split in half, with one half (which makes up two-thirds of the dataset) being used just for testing and the other half (which makes up the last one-third) being utilized for learning. Two-thirds of the

data from the first batch have likewise been divided into two halves. The SVM classifier's pure learning process took up the first half, while the fitness function's calculation took up the second half (the validation of the SVM model). Only the last third of the data was used for the trained classifiers' testing. Using a bone marrow picture as a starting point, a support vector machine (SVM) and a genetic algorithm (GA) were used to identify blood cells. In comparison to the most effective feature selection strategy, we improved blood cell identification accuracy by more than 25% (relatively) (linear SVM ranking). Nasir et al. [34] observed a total of 500 pictures using the Leica microscope. Out of these, 200 ALL and 300 AML were taken from acute leukemia blood samples. They have employed the Simplified Fuzzy ARTMAP (SFAM) and Multilayer Perceptron (MLP) neural networks. The MLP network has been trained using Bayesian Regulation and Levenberg–Marquardt algorithms. The Bayesian Regulation algorithm-trained MLP network delivered the best classification results, with testing accuracy for all suggested features of 95.70%.

There are two different kinds of datasets [35]. Both the ALL-IDB1 and ALL-IDB2 have segmented WBCs to test the categorization of blast cells, and both may be used to evaluate the segmentation capabilities of algorithms, classification systems, and methods for image pre-processing algorithms, classification systems, and methods for image pre-processing. In the research, the closest neighbor and support vector machine (SVM) concepts are discussed. Leukemia was detected using suggested characteristics, and the KNN classifier identified it with an overall accuracy of 93%. Furey et al. [36] collected a dataset of 72 patients. Of those, 24 were utilized for testing and the remaining 38 were used for training. SVM algorithm was considered for the algorithmic interpretation. Through SVM, an accuracy percentage of 70% is attained. Su-In Lee et al. [37] observed 30 patient gene expression samples. They were examined using two different datasets. Then they have applied Multiple regression techniques, the MERGE algorithm, and leave-one-out cross validation. MERGE algorithm had an accuracy of 83%, LOOCV had an accuracy of 72%, and multiple regression techniques had an accuracy of just 60%.

Nayana B. Sen et al. [38] used the dataset from American Society of Hematology online image bank. Following feature extraction for a few chosen characteristics, picture segmentation was done to increase quality and obtain key parts of the images. K closest neighbor technique was then employed for classification. Healthy and malignant cells have Harsdorf Dimensions of 1.5501 and 1.7828, respectively. In terms of specificity and precision, it was discovered that KNN classifier is about as well as SVM classifier.

3.2 Diagnosis by Using ANN and CNN Algorithms

In the work of Rana Zeeshan Haider et al. [39], artificial neural network (ANN) with principal component analysis (PCA) have been used. Their dataset consists of 1067 patients. In this dataset there were 44 patients of APML (PML-RARA), 181 of

AML (excluded APML), 89 of chronic myeloid leukemia (CML), 51 of myelodysplastic syndrome (MDS), 71 of myeloproliferative disorders (MPN) except CML, 10 of MDS/MPN, 136 of acute lymphocytic leukemia (ALL), 9 of Hodgkin's lymphoma (HL), 95 of non-Hodgkin's lymphoma (NHL), 32 of multiple myeloma, and 349 of normal control were present. At Pakistan's National Institute of Blood Disease and Bone Marrow Transplantation (NIBD and BMT), they were prospectively enrolled. In conjunction with artificial neural networks, principal component analysis (PCA) was performed (ANN). The accuracy rates of the ANN for the training and testing datasets were 95.7 and 97.7%, respectively. Rana Zeeshan Haider et al. [39] in their study covered a diagnosis of 1067 patients. Hematological neoplasms were the cause of their pain. Principal component analysis (PCA) and artificial neural network (ANN) predictive modeling are both employed. For the training and testing datasets, respectively, the ANN model was determined to be sufficient with values of 95.7 and 97.7%. For the purpose of analyzing, a dataset of 9395 images is used by J. Rodellar et al. [30]. This sample size was collected from blood smears of 218 patients. Support vector machines (SVM), decision trees, and neural networks are employed as approaches. The mean value SVM classifier's overall accuracy is 88.3%, however the accuracy of the other approaches is less than that of the SVM.

The value of fuzzy logic-based systems in the detection of different illnesses is highlighted by Nega Gupta et al. in their study [40]. These technologies serve as assisting tools for medical personnel and can be extremely helpful in developing regions with low doctor-to-patient ratios. The article also looks at various fuzzy models used in healthcare systems for decision-making, with a focus on their applications in the diagnosis of a number of illnesses, such as Alzheimer's, Chronic Kidney Disease, Cholera, Hepatitis B, Coronary Artery Disease, Diabetes, Asthma, COVID-19, Stroke, Renal Cancer, and Heart Disease. In order to highlight the value of fuzzy logic-based systems in the healthcare industry, the chapter presents the findings from many researchers in the identification of these disorders.

To prepare the dataset, the researchers used the sources of ALL-IDB and ASH Image bank. Nizar Ahmed et al. [41] have done data augmentation to upsurge dataset size and to circumvent memorization. They used the Convolutional Neural Network model (CNN). Feature extraction which could be done automatically is performed for the images. 25 epochs and 32 batch sizes were used in the model's training. The dataset size was increased by using image modifications. Experiments show that the CNN model works with 81.74% accuracy when categorizing all subtypes into different groups and 88.25% accuracy when comparing leukemia to healthy persons. Compared to other well-known machine learning algorithms, the CNN model performs better.

ALL-Image Database (IDB) is used as data source to frame the dataset by Sarmad Shafique et al. [42] in their work. Besides this data source, researchers have used Google too, to increase the size of the dataset. From Google they collected 50 microscopic blood images. These samples were validated by the expert oncologist. Then, AlexNet, a pretrained Convolutional Neural Networks, for detection of ALL and classification of its sub types is used. For the architecture of deep neural

networks, transfer learning was applied. Data augmentation was used to diagnose leukemia with a 99.50% accuracy rate and classify its subtypes with a 96.06% accuracy rate. To find the most effective deep learning architecture for categorization, researchers should use a variety of deep learning designs. T. T. P. Thanh et al. [43] have used the data source of ALLIDB for building their database of images. They performed several types of transformations on the images to enhance the size of the dataset. By doing this a database of 1188 images were built. Seven-layer convolutional neural network was employed. The first five levels extract features, while the next two layers categorize the features. The percentage of photos used for testing was 30%. A 96.60% accuracy rate was achieved. In conclusion, CNN is quite trustworthy for spotting blood cancer in its early stages.

MalekMalek ADJOUADI et al. [44] collected the samples from several hospitals. The samples consist of normal as well as infected cells. Further data analysis was performed in order to extract just the cells that satisfied the relevant criteria. Prior to categorization, only five features underwent feature extraction. ALL and AML sample categorization was carried out using ANN. Three testing and training cycles were carried out for the ALL classification. Testing accuracy for the largest number of samples was 98.46. Only one test was carried out for AML categorization. It provided 97.27% accuracy. ANNs may be trained to distinguish between AML and ALL by utilizing fewer parameters.

SBILab was the source of dataset taken by Sara Hosseinzade h Kassani et al. [45]. The dataset is based on the differentiation between benign and malignant cells in microscopic pictures of B-ALL white blood cancer. 76 different people make up the dataset, which has an overall cell picture of 7272 ALL and 3389 normal cells. Preprocessing of the data increased the number of photos. It is suggested to use a hybrid CNN model, which incorporates low-level characteristics from intermediate layers. The CNN MobileNet and VGG16 architectures were employed. At the classifier layer, two output neurons with SoftMax non-linearity activation functions are employed to associate normal and malignant cases. 967 test photos were used to get the results. The suggested model produced an accuracy of 96.17%. Results obtained indicate that merging characteristics discovered by deep models enhances performance and produces more precise results.

In the work of Dr. BBM Krishna et al. [46] dataset is a compilation of Golub's published expression measurements. From 72 individuals with ALL or AML, profiles have been created. Using dimensionality reduction approaches for classification, such as Signal-to-Noise Ratio, Class-Separability, etc., the best genes for classification were found. A fuzzy hypersphere neural network classifier was used for categorization. Two genes were all that was required to achieve high accuracy. With the ALL/AML dataset, the FHSNN classifier trained and tested on average substantially quicker than more established techniques like KNN and SVM. FHSNN produced 100% accuracy compared to 97.1% for conventional approaches. Dhvani Kansara et al. [47] in their work used 12,500 images in their dataset. There were 3000 enhanced photos of each kind (Eosinophil, Lymphocyte, Monocyte, and Neutrophil). 410 of the photos were unaltered originals, classifying photos into different cell types using a deep convolutional neural network. In the beginning, many

layers of convolution and pooling are applied to extract every feature that could be present. A collection of WBC pictures with various orientations were produced using image modifications. Because of back-propagation, the model's accuracy increased with the number of epochs. After 30 epochs, the validation set accuracy remained almost unchanged. Precision is measured to be 83%. Recall and F1 score were both 78%.

Gulshan Sharma et al. [48] prepared a dataset of 364 images. This is a public dataset having colored microscopic images of WBC. From the data, a training set of 80% was used. Approximately 10% of the data were used for validation. Test set was created using the remaining 10%. Five 2D convolutional layer convolutional neural networks are employed. The most epochs that could be specified was 10. High rates of accuracy were seen. Accuracy for binary classification was 99.76%. Accuracy for multiple categorization was 98.14%.

Gayathri S et al. [49] in their work used a dataset of tiny images. The complete dataset is made up of microscopic images that were taken in the lab using a Canon Power Shot G5 camera. After acquisition, segmentation and cleaning were carried out. First, an ANN was put into action. Its performance was contrasted with that of the SVM. CNN is also used to implement the proposed tasks. The feature extraction method-based recognition system has an efficiency of 89.47% with SVM and 92.10% with ANN. The effectiveness of the CNN-based feature extraction approach is 93%. TTP et al. [13] worked using CNN for the preparations of clinical DSS. They had a database of 80 samples out of which 40 are of normal cells whereas, another 40 samples are of abnormal cells. This database was used for training the model. Another dataset has 28 samples out of which 19 were for normal cells and 9 were of abnormal cells. This dataset was used for testing the model. A unique approach of classifying acute leukemia uses convolution neural networks (CNN). Convolutional, pooling, and fully linked layers are only a few of the many layers that make up a CNN. Pooling layers lower the dimensionality of the data by averaging the output of convolutional layers, whereas convolutional layers utilize filters to extract features from the input data. The fully connected layers then use this processed data to make a prediction or classification. It gives a very good performance in the classification procedure, achieving 96.43% accuracy to distinguish between normal and pathological cell pictures from the provided database. This data collection comes in two separate [42] iterations. Acute Lymphoblastic Leukemia-IDB 1 had 108 images, 49 of which were of leukemia patients and 59 of which were of healthy individuals. For the acute lymphoblastic leukemia-IDB 2 investigation, 260 single-cell pictures were utilized to collect the data, of which 130 were collected from leukemia patients and 130 from healthy people. A data augmentation strategy was utilized to reduce over training. Convolutional neural networks (CNN), SVM (support vector machines), and DCCN (deep convolutional neural network) were then applied. The categorization of the subtype of acute lymphoblastic leukemia achieved sensitivity of 96.74%, specificity of 99.03%, and accuracy of 96.06%. It was 100% sensitive, 98.11% specific, and 99.50% accurate at detecting acute lymphoblastic leukemia.

Saeid Afshar et al. [50] observed from the medical records of 131 individuals (63 of whom had cancer and the remainder did not) with pathological results, 41 clinical

and laboratory parameters were chosen. They performed the Artificial Neural Network (ANN), LM algorithm interpretations on the dataset. The learning effectiveness was 0.094. The area under the ROC curve was 0.967, and there was a good correlation between the output of the trained network for the test data and the actual results of the test data. Consequently, artificial neural networks may be used to recognize leukemia quickly and accurately. Theera-Umpon et al. [51] analyzed the clustering of patch-based blood. The six kinds of white blood cells represented in the dataset are myeloblast, promyelocyte, myelocyte, metamyelocyte, band, and PMN. Each of the six cell classes has a corresponding number of hand segmented images: 20, 9, 139, 33, 45, and 185. Fuzzy C-means clustering and mathematical morphology are the foundations of the segmentation. In comparison to an expert's ground truth, we achieve a decent segmentation and promising classification results. The suggested patch-based segmentation approach makes more sense than the pixel-based segmentation techniques because of the inconsistent grayscale in each area of a white blood cell picture.

Leyza Baldo et al. [52] tested over 100 photos in grayscale using images from CellAtlas.com. To increase the accuracy of segmentation, employ basic morphological operators and investigate the scale-space characteristics of a toggling operator. Future research, such as the categorization of WBC using shape descriptors taken from segmented nuclei, is encouraged by the accurate nucleus segmentation results.

3.3 Diagnosis by Using Random Forest and Decision Tree Algorithms

Liyan Pan et al. [53] have used a dataset of 661 children. They were of the ages less than 16 years. All of them were diagnosed recently suffering from ALL. The researchers used Random Forest (RF) and Decision Tree (DT) for the predictive analysis. The prediction given by RF presented an accuracy of 0.831 specificity as 0.895, PPV as 0.880, and AUC as 0.902 as compared to DT in 4 of 6 measurements. The results given by RF stand out as compared to DT.

Jakkrich Laosai et al. [54] have used a sample size of 500. Out of these 500 samples, 200 have been used for training and the rest of the 300 have been used for testing. These samples were of acute-leukemia. The cluster of differentiation (CD) marker is used. The testing result shows an accuracy of 99.67%.

The total of 76 sample size is used by Gonzalez Jesus A [55] et al. in their work. Out of this, 56 samples were used for training the dataset. The rest of the 20 are used for testing. SVM, ANN, RF, k-Means are used. The difference between the acute myeloblastic and lymphoblastic leukemia relations was achieved with an accuracy of 95.5% from SVM, 79.4% from ANN, 82.3% from RF, and 82.6% from k-Means. Tatdow Pansombut et al. [56] prepared two datasets. Dataset 1 consists of 93 samples. These samples are of WBCs. The second collection consists of pre-T and

pre-B cells from the ASH image library, representing ALL subtypes. The classification process uses Convent, a convolutional neural network. Automatic feature extraction was carried out. 46 features were chosen for feature extraction for SVM-GA, MLP, and Random Forest. GA was used to inform the procedures for parameter and feature optimization. The average accuracy rates for ConVNet, SVM-GA, MLP, and random forest were, respectively, 81.74%, 81.65%, 76.12%, and 78.43%. In each of the three classes, CNN outperformed MLP and random forest. Haneen et al. [7] worked for the diagnosis of leukemia by using ML applications. The search algorithms made use of Boolean logic and MeSH terminology, such as the phrases "Leukemia" and "Leukemia, Myeloid," as well as terms related to AHI procedures. According on the type of leukemia, the studies were divided in these categories: ALL (13), AML (8), CLL (3), and CML (1). For both AML and ALL, two studies offered diagnostic models. ALL Image DataBase (IDB), a popular digital resource, was utilized in 5 investigations (42%).

There are two datasets for 42 ALL-IDB. While the cells in dataset (2) are segmented, those in dataset (1) are not, providing for exercises in both segmentation and classification. Deep learning and machine learning (ML) methods including SVM, KNN, RF, LR, RC, and CNN are employed. Pattern recognition-based segmentation algorithms (such as fuzzy c-means and k-means) were most often used, followed by threshold-based algorithms (e.g., watershed). The accuracy of the methods employed for ALL varied from 74% to 99.5%. Using the SVM method, ALL detection accuracy was 74%. The accuracy of the algorithms utilized in AML has ranged from 82% to 97%. After utilizing many techniques, including Bayesian clustering (BC), a diagnosis of CLL using a flow cytometer was made with 99.6% accuracy.

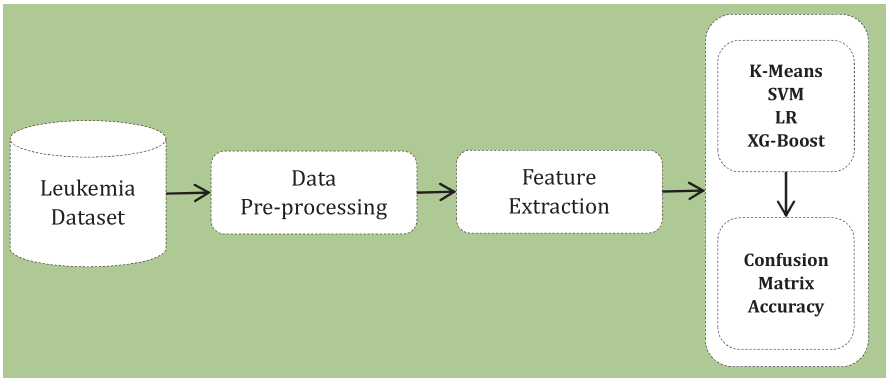


Fig. 2 The proposed method

4 Results and Discussion

The data was obtained from GitHub, Fig. 2 depicts a confusion matrix that is used to demonstrate how well a classification system works. A confusion matrix shows and summarizes a categorization method's efficacy. Five important algorithms and classifiers are evaluated on the dataset for performance and accuracy.

The algorithms and classifiers on which the accuracy is checked are (i) K-Means, (ii) Naive Bayes, (iii) Support Vector Means, (iv) Logistic Regression, and (v) XG-Boost algorithm. The confusion matrix for all above-mentioned five algorithms have been prepared and depicted below.

4.1 K-Means

The confusion matrix for K-Means is shown in Fig. 3. The accuracy of this algorithm when trained by the dataset came out to be 0.765.

4.2 Naive Bayes

0.912 is the accuracy of the Naive Bayes algorithm when it is trained by the dataset. Its confusion matrix is depicted in Fig. 4.

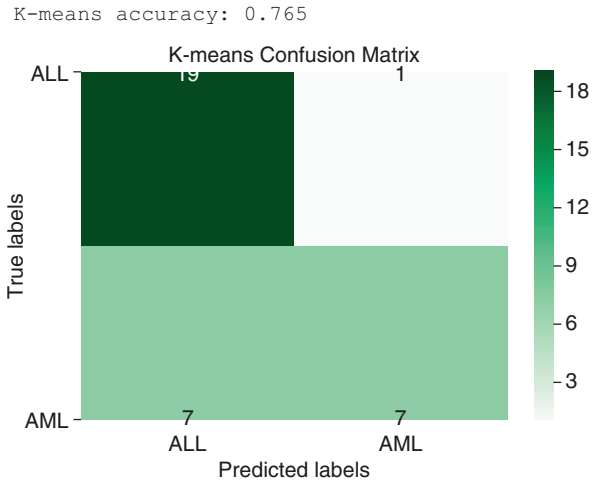


Fig. 3 K-means confusion matrix

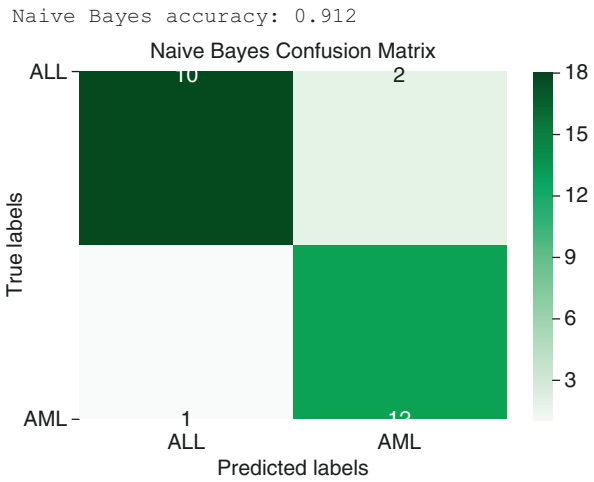


Fig. 4 Naive Bayes confusion matrix

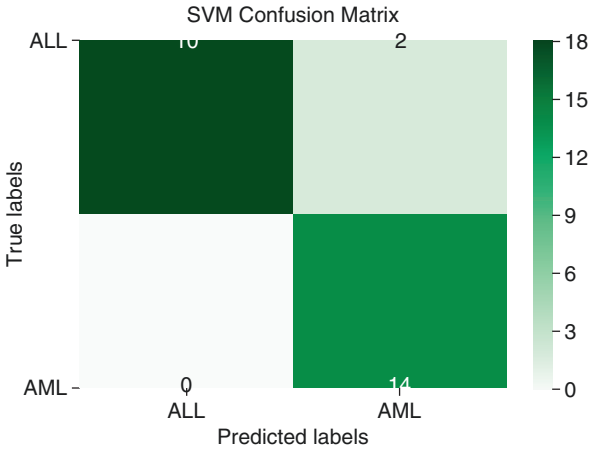


Fig. 5 Support vector means confusion matrix

4.3 Support Vector Means

The confusion matrix for Support Vector Means is shown in Fig. 5. The accuracy of this algorithm when trained by the dataset came out to be 0.941.

4.4 Logistic Regression

1.0 is the accuracy of Logistic Regression algorithm when it is trained by the dataset. Its confusion matrix is depicted in Fig. 6.

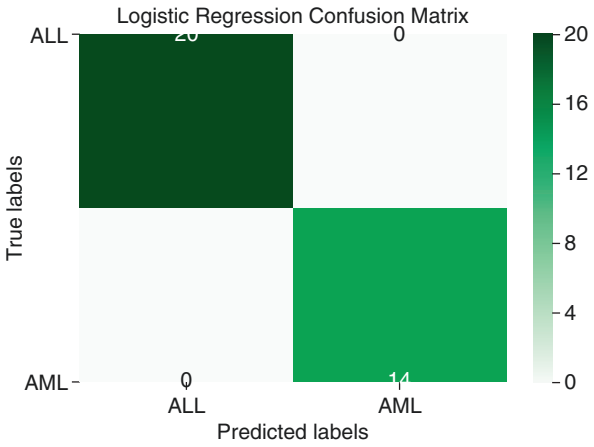


Fig. 6 Logistic regression confusion matrix

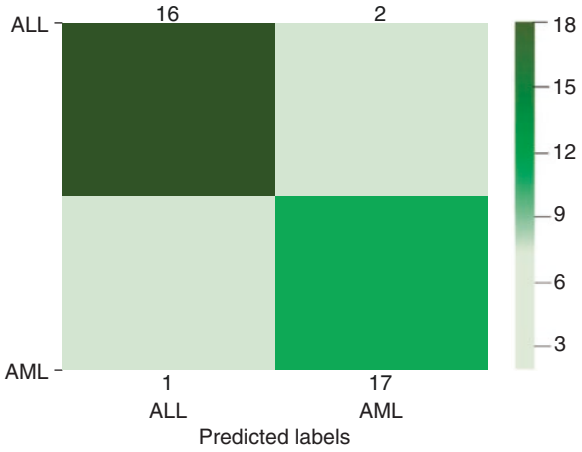


Fig. 7 XG-Boost confusion matrix

4.5 XG-Boost

The confusion matrix for XG-Boost is pictured in Fig. 7. The accuracy of this algorithm when trained by the dataset came out to be 0.912.

4.6 Accuracy Achieved in Algorithms

Various algorithms like K-means, Naive Bayes, SVM, Logistic Regression, and XG-Boost are trained by the dataset. The accuracy is measured on all the five algorithms. The resultant accuracy is shown in Fig. 8. It shows that logistic regression resulted in 100% accuracy rate.

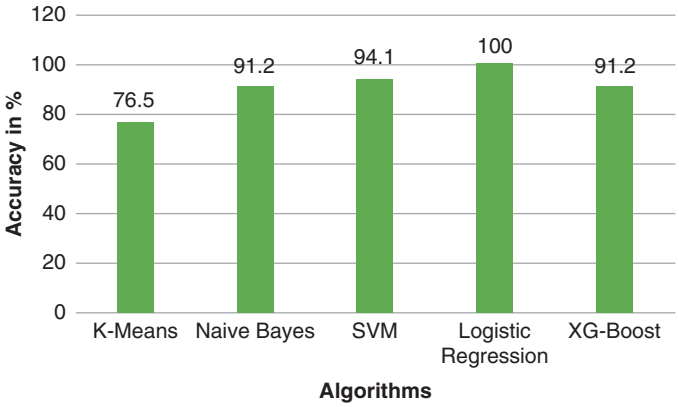


Fig. 8 Accuracy achieved in algorithms

5 Conclusion

In this chapter a thorough examination of numerous research articles is done that used SVM, KNN, Naive Bayes, K-Means, logistic regression, and various artificial intelligence algorithms like ANN and CNN for the diagnosis, detection, and classification of the leukemia disease. The accuracy of K-Means, Naive Bayes, Support Vector Means, Logistic Regression, and XG-Boost algorithm are checked by training them using the dataset. The accuracy of K-Means is 0.765, Naive Bayes is 0.912, Support vector means is 0.941, Logistic Regression is 1.0, and XG-Boost algorithm is 0.912, respectively. The result shows that logistic regression is the best approach for the dataset used in this work. The comparative analysis among various algorithms also shows that logistic regression gives best accuracy. The leukemia patients can be detected using this algorithm and also could be classified as per the level of suffering, that is, acute or chronic. Leukemia being fatal, cancer needs to be rectified at the earliest. As it has been shown in this chapter, logistic regression gives the best accuracy, hence use of logistic regression algorithms can help the medicos in identifying the right type of leukemia at the earliest stage.

6 Declarations

- Competing interests: The authors have none that are materially material, either financially or otherwise.
- Funding: The authors affirm that they did not receive any money, grants, or other forms of assistance for the creation of this publication.
- Ethics approval: It is an observational research. There is no requirement for ethical clearance.

- Consent to participate: Each and every person who was a part of the study gave their informed consent.
- Consent to publish: According to the authors, the full text of the article was published with the participants' informed consent.
- Author information: Information about the writers is not mentioned in the text at all.

References

1. Pal, S. K., Bandyopadhyay, S., & Ray, S. S. (2006). Evolutionary computation in bioinformatics: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 36*(5), 601–615.
2. Patel, H., & Rajput, D. (2011). Data mining applications in present scenario: A review. *International Journal of Soft Computing, 6*(4), 136–142.
3. Angermueller, C., Parnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology, 12*(7), 878.
4. Jones, W., Alasoo, K., Fishman, D., & Parts, L. (2017). Computational biology: Deep learning. *Emerging Topics in Life Sciences, 1*(3), 257–274.
5. Bhaskar, H., Hoyle, D. C., & Singh, S. (2006). Machine learning in bioinformatics: A brief survey and recommendations for practitioners. *Computers in Biology and Medicine, 36*(10), 1104–1125.
6. Pui, C.-H., Robison, L. L., & Look, A. T. (2008). Acute lymphoblastic leukaemia. *The Lancet, 371*(9617), 1030–1043.
7. Salah, H. T., Muhsen, I. N., Salama, M. E., Owaidah, T., & Hashmi, S. K. (2019). Machine learning applications in the diagnosis of leukemia: Current trends and future directions. *International Journal of Laboratory Hematology, 41*(6), 717–725.
8. Srisukkham, W., Zhang, L., Neoh, S. C., Todryk, S., & Lim, C. P. (2017). Intelligent leukaemia diagnosis with bare-bones PSO based feature optimization. *Applied Soft Computing, 56*, 405–419.
9. Dohner, H., Weisdorf, D. J., & Bloomfield, C. D. (2015). Acute myeloid leukemia. *New England Journal of Medicine, 373*(12), 1136–1152.
10. O'Donnell, M. R., Abboud, C. N., Altman, J., Appelbaum, F. R., Arber, D. A., Attar, E., Borate, U., Coutre, S. E., Damon, L. E., Goorha, S., et al. (2012). Acute myeloid leukemia. *Journal of the National Comprehensive Cancer Network, 10*(8), 984–1021.
11. Patel, N., & Mishra, A. (2015). Automated leukaemia detection using microscopic images. *Procedia Computer Science, 58*, 635–642.
12. Shafique, S., & Tehsin, S. (2018). Computer-aided diagnosis of acute lymphoblastic leukaemia. *Computational and Mathematical Methods in Medicine, 2018*.
13. TTP, T., Pham, G. N., Park, J. -H., Moon, K. -S., Lee, S. -H., Kwon, K. R., et al. (2017). Acute leukemia classification using convolution neural network in clinical decision support system. In *CS & IT conference proceedings* (vol. 7).
14. Thakur, T., Batra, I., Luthra, M., Vimal, S., Dhiman, G., Malik, A., & Shabaz, M. (2021). Gene expression-assisted cancer prediction techniques. *Journal of Healthcare Engineering, 2021*.
15. Waseem, M. H., Nadeem, M. S. A., Abbas, A., Shaheen, A., Aziz, W., Anjum, A., Manzoor, U., Balubaid, M. A., & Shim, S.-O. (2019). On the feature selection methods and reject option classifiers for robust cancer prediction. *IEEE Access, 7*, 141072–141082.
16. Faivdullah, L., Azahar, F., Htike, Z. Z., & Naing, W. (2015). Leukemia detection from blood smears. *Journal of Medical and Bioengineering, 4*(6).

17. Rawat, J., Singh, A., Bhadauria, H., & Virmani, J. (2015). Computer aided diagnostic system for detection of leukemia using microscopic images. *Procedia Computer Science, 70*, 748–756.
18. Negm, A. S., Hassan, O. A., & Kandil, A. H. (2018). A decision support system for acute leukaemia classification based on digital microscopic images. *Alexandria Engineering Journal, 57*(4), 2319–2332.
19. Abdeldaim, A. M., Sahlol, A. T., Elhoseny, M., & Hassanien, A. E. (2018). Computeraided acute lymphoblastic leukemia diagnosis system based on image analysis. In *Advances in soft computing and machine learning in image processing* (pp. 131–147). Springer.
20. Prinyakupt, J., & Pluempitiwiriyawej, C. (2015). Segmentation of white blood cells and comparison of cell morphology by linear and na¨ıve Bayes classifiers. *Biomedical Engineering Online, 14*(1), 1–19.
21. Netto, O. P., Nozawa, S. R., Mitrowsky, R. A. R., Macedo, A. A., Baranauskas, J. A., & Lins, C. (2010). Applying decision trees to gene expression data from DNA microarrays: A leukemia case study. In *XXX congress of the Brazilian computer society, X workshop on medical informatics* (p. 10). Belo Horizonte MG.
22. Rehman, A., Abbas, N., Saba, T., Rahman, S. I. U., Mehmood, Z., & Kolivand, H. (2018). Classification of acute lymphoblastic leukemia using deep learning. *Microscopy Research and Technique, 81*(11), 1310–1317.
23. Cernea, A., Fernández-Martínez, J. L., de Andrés-Galiana, E. J., Galván Hernández, J. A., García Pravia, C., & Zhang, J. (2018). Analysis of clinical prognostic variables for triple negative breast cancer histological grading and lymph node metastasis. *Journal of Medical Informatics and Decision Making, 1*(1), 14–36.
24. Candia, J., Cherukuri, S., Guo, Y., Doshi, K. A., Banavar, J. R., Civin, C. I., & Losert, W. (2015). Uncovering low-dimensional, mir-based signatures of acute myeloid and lymphoblastic leukemias with a machine-learning driven network approach. *Convergent Science Physical Oncology, 1*(2), 025002.
25. Ni, W., Tong, X., Qian, W., Jin, J., & Zhao, H. (2013). Discrimination of malignant neutrophils of chronic myelogenous leukemia from normal neutrophils by support vector machine. *Computers in Biology and Medicine, 43*(9), 1192–1195.
26. Paswan, S., & Rathore, Y. K. (2017). Detection and classification of blood cancer from microscopic cell images using SVM KNN and NN classifier. *International Journal of Advance Research, Ideas and Innovations in Technology, 3*, 315–324.
27. Amin, M. M., Kermani, S., Talebi, A., & Oghli, M. G. (2015). Recognition of acute lymphoblastic leukemia cells in microscopic images using k-means clustering and support vector machine classifier. *Journal of Medical Signals and Sensors, 5*(1), 49.
28. Kumar, S., Mishra, S., Asthana, P., et al. (2018). Automated detection of acute leukemia using k-mean clustering algorithm. In *Advances in computer and computational sciences* (pp. 655–670). Springer.
29. Daqqa, K. A. A., Maghari, A. Y., & Al Sarraj, W. F. (2017). Prediction and diagnosis of leukemia using classification algorithms. In *2017 8th International Conference on Information Technology (ICIT)* (pp. 638–643). IEEE.
30. Rodellar, J., Alferez, S., Acevedo, A., Molina, A., & Merino, A. (2018). Image processing and machine learning in the morphological analysis of blood cells. *International Journal of Laboratory Hematology, 40*, 46–53.
31. Joshi, M. D., Karode, A. H., & Suralkar, S. (2013). White blood cells segmentation and classification to detect acute leukemia. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2*(3), 147–151.
32. Kazemi, F., Najafabadi, T. A., & Araabi, B. N. (2016). Automatic recognition of acute myelogenous leukemia in blood microscopic images using k-means clustering and support vector machine. *Journal of Medical Signals and Sensors, 6*(3), 183.
33. Osowski, S., Siroic, R., Markiewicz, T., & Siwek, K. (2008). Application of support vector machine and genetic algorithm for improved blood cell recognition. *IEEE Transactions on Instrumentation and Measurement, 58*(7), 2159–2168.

34. Nasir, A. A., Mashor, M. Y., & Hassan, R. (2013). Classification of acute leukaemia cells using multilayer perceptron and simplified fuzzy artmap neural networks. *The International Arab Journal of Information Technology, 10*(4), 1–9.
35. Chatap, N., & Shibu, S. (2014). Analysis of blood samples for counting leukemia cells using support vector machine and nearest neighbour. *IOSR Journal of Computer Engineering (IOSR-JCE), 16*(5), 79–87.
36. Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics, 16*(10), 906–914.
37. Lee, S.-I., Celik, S., Logsdon, B. A., Lundberg, S. M., Martins, T. J., Oehler, V. G., Estey, E. H., Miller, C. P., Chien, S., Dai, J., et al. (2018). A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nature Communications, 9*(1), 1–13.
38. Sen, N. B., & Mathew, M. (2016). Automated AML detection from complete blood smear image using KNN classifier. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 5*(7).
39. Haider, R. Z., Ujjan, I. U., & Shamsi, T. S. (2020). Cell population data–driven acute promyelocytic leukemia flagging through artificial neural network predictive modeling. *Translational Oncology, 13*(1), 11–16.
40. Gupta, N., Singh, H., & Singla, J. (2022). Fuzzy logic-based systems for medical diagnosis – A review. In *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1058–1062). IEEE.
41. Ahmed, N., Yigit, A., Isik, Z., & Alpkocak, A. (2019). Identification of leukemia subtypes from microscopic images using convolutional neural network. *Diagnostics, 9*(3), 104.
42. Shafique, S., & Tehsin, S. (2018). Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks. *Technology in Cancer Research & Treatment, 17*, 1533033818802789.
43. Thanh, T., Vununu, C., Atoev, S., Lee, S.-H., & Kwon, K.-R. (2018). Leukemia blood cell image classification using convolutional neural network. *International Journal of Computer Theory and Engineering, 10*(2), 54–58.
44. Adjouadi, M., Ayala, M., Cabrerizo, M., Zong, N., Lizarraga, G., & Rossman, M. (2010). Classification of leukemia blood samples using neural networks. *Annals of Biomedical Engineering, 38*(4), 1473–1482.
45. Kassani, S. H., Kassani, P. H., Wesolowski, M. J., Schneider, K. A., & Deters, R. (2019). A hybrid deep learning architecture for leukemic b-lymphoblast classification. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 271–276). IEEE.
46. Kanth, B. K. (n.d.). A fuzzy-neural approach for leukemia cancer classification.
47. Kansara, D., Sompura, S., Momin, S., & D'Silva, M. (2018). Classification of WBC for blood cancer diagnosis using deep convolutional neural networks. *International Journal of Research in Advent Technology, 6*(12), 3576–3581.
48. Sharma, G., & Kumar, R. (2019). Classifying white blood cells in blood smear images using a convolutional neural network. *International Journal of Innovative Technology and Exploring Engineering, 8*(9S), 103–108.
49. Gayathri, S., & Jyothi, R. (2018). An automated leucocyte classification for leukemia detection. *International Research Journal of Engineering and Technology, 5*(5), 4254–4264.
50. Afshar, S., Abdolrahmani, F., Vakili, T. F., Zohdi, S. M., & Taheri, K. (2011). Recognition and prediction of leukemia with artificial neural network (ANN).
51. Theera-Umpon, N. (2005). Patch-based white blood cell nucleus segmentation using fuzzy clustering. *ECTI-EEC, 3*(1), 15–19.
52. Dorini, L. B., Minetto, R., & Leite, N. J. (2007). White blood cell segmentation using morphological operators and scale-space analysis. In *XX Brazilian symposium on computer graphics and image processing (SIBGRAPI 2007)* (pp. 294–304). IEEE.

53. Pan, L., Liu, G., Lin, F., Zhong, S., Xia, H., Sun, X., & Liang, H. (2017). Machine learning applications for prediction of relapse in childhood acute lymphoblastic leukemia. *Scientific Reports, 7*(1), 1–9.
54. Laosai, J., & Chamnongthai, K. (2018). Classification of acute leukemia using medical-knowledge-based morphology and cd marker. *Biomedical Signal Processing and Control, 44*, 127–137.
55. Gonzalez, J. A., Olmos, I., Altamirano, L., Morales, B. A., Reta, C., Galindo, M. C., Alonso, J. E., & Lobato, R. (2011). Leukemia identification from bone marrow cells images using a machine vision and data mining strategy. *Intelligent Data Analysis, 15*(3), 443–462.
56. Pansombut, T., Wikaisuksakul, S., Khongkraphan, K., & Phon-On, A. (2019). Convolutional neural networks for recognition of lymphoblast cell images. *Computational Intelligence and Neuroscience, 2019*.

A Review on Mode Collapse Reducing GANs with GAN's Algorithm and Theory

Shivani Tomar and Ankit Gupta

1 Introduction

The generative adversarial networks (GANs) have turned into a very trending topic of discussion due to its popularity and have wide range applications. They work very well on images, audio, and video [13–16]. They have many applications in the medical field also which includes new drug discovery [21], detection of disease, etc.

GAN is made up of two networks which are discriminator and generator. It works on the principle of min-max rule. The objective function is to maximize the loss equation for the discriminator and minimize it for the generator. The generator tries to capture the probability distribution from the dataset to generate new examples of data whereas discriminator is a binary classifier which tries to recognize the true dataset over the data generated by the generator. The adversarial idea works very well and gives excellent results. But, due to GAN's complex structure, it is tough to train GAN and faces a lot of issues like non-convergence, instability, and mode collapse in which mode collapse is a very big issue because it blocks the true potential of the generative adversarial network by stopping it from generating more diverse data. Mode collapse is not the problem of only vanilla GAN, but this is also seen in many GAN methods which solve the problems using the GAN concept. MOL-GAN is a GAN method which was introduced to solve chemical synthesis problems by generating direct molecular graphs of elements. This model also suffered from mode collapse which was avoided by using improved Wasserstein GAN and mini-batch discriminator [21]. EEG-GAN is used to generate electroencephalographic (EEG) brain signals and suffers from mode collapse and it is avoided using WGAN [28]. BAGAN (Balancing GAN) is used as an augmentation framework for

S. Tomar (✉) · A. Gupta
Department of Computer Engineering, J.C. Bose University of Science & Technology, YMCA, Faridabad, Haryana, India
e-mail: shivanitomar1207@gmail.com; ankitgupta24101@gmail.com

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_2

balancing the imbalanced data by generating minority class data and this framework must be free from mode collapse to generate diverse data to balance the dataset. To do the same, regularization is done along with the encoder coupled with BAGAN [29]. Mode collapse problem is also faced by text generation using adversarial networks and this problem is resolved by inverse reinforcement learning [13]. Self-Attention GAN uses spectral regularization for smoothing the training process and diversity of image generation [17].

There are many applications which are using adversarial networks and mostly face mode collapse problems and are resolved by many different ways. So, we can say that mode collapse is the big stone in the GAN (generative adversarial networks).

2 Literature Survey

GAN is the generative model which falls under the method of semi unsupervised learning. In unsupervised learning methods, there are no labels on the dataset and the learning is done only through the data itself. All clustering algorithms like k-means, k-median algorithms, feature extraction algorithms like PCA (Principal Component Analysis), etc., come under unsupervised learning methods. In supervised learning, labels are there along with the data in the dataset. Classification, regression, segmentation, etc., are the supervised learning methods. In GAN, discriminator works on supervised learning method and generator works on unsupervised learning method.

Generative Models Models which are used to generate data are called generative models. Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM), etc., are examples of generative models. A generative model captures the joint probability P(x/y), that is, what is the probability of occurring event x while event y already occurs if the condition is: given. If there is no such event then it calculates only P(x), that is, probability of occurring event x. Most of the generative models work on the maximum likelihood function.

Figure 1 shows the structure of generative models on the basis of maximum likelihood. On the basis of maximum likelihood function, generative models are divided into two categories – the first one is known as explicit density generative models and another one is known as implicit density generative models. The former one requires prior probability distribution of the data to calculate posterior along with the maximum likelihood estimation while implicit density model does not require prior probability distribution to calculate the probability distribution of data. Explicit density models are further classified into tractable and intractable (approximate) density models. As their name suggests, in tractable models, function is computable while in approximate density model function is not computable and derived by using other ways. Pixel Recurrent Neural Networks, Fully Visible Belief Networks (FVBN), Neural Autoregressive Distribution Estimation (NADE), Masked Autoencoder for Distribution Estimation (MADE), etc., are the examples of

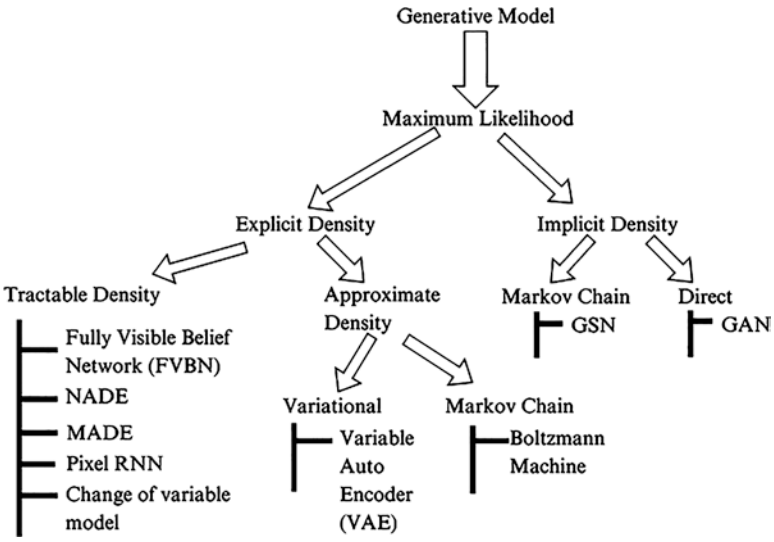


Fig. 1 Hierarchical structure of generative models on the basis of maximum likelihood

tractable density models while approximate density is further classified into two categories which are variational and Markov chain. Variable auto encoder comes under variational density and Boltzmann machine is categorized under Markov chain. In an implicit density model, the probability distribution is calculated on the basis of Monte Carlo Markov chain method or distribution is directly calculated. GSN (deep generative stochastic network) is an example of Markov chain implicit density method.

GAN comes under the implicit density model in the direct probability distribution method as it does not require any external probability distribution to calculate joint probability.

Maximum likelihood based generative models have some disadvantages like – tractable density models are very slow as they work sequentially. For example, FVBNs take two minutes to generate one second of audio. Variational autoencoders (VAEs) are hard to optimize and Markov chains are very slow to converge.

To avoid these disadvantages, GAN was designed which has the following advantages over other generative models:

GAN does not use runtime, proportional to dimension of the data, and it can generate samples in parallel while other generator neural networks like pixel RNN generate data sequentially.

In GAN, there is no variational lower bound required while VAE requires lower bound.

They are instinctively regarded as generating better results as compared to the other methods while others like VAE compromise with the image quality.

With these advantages, GAN comes along with some disadvantages which are the issues faced by GAN while training them [25].

Idea Behind GAN Most of the generative models use L2 loss function which is $|x_i - x_j|^2$ while GAN works on the adversarial idea for generating data samples, that is, both generator and discriminator trying to oppose each other. Generator tries to generate data samples in such a way that discriminator might not be able to recognize which one is fake and which one is real and discriminator always tries to reject the data generated by the generator:

$$\arg\min_G \max_D V(D,G) = E_x p_{\text{data}}(x)\left[\log\left(D(x)\right)\right] + E_{zp_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

The objective function of the generator is to minimize Eq. (1) and discriminator tries to maximize the Eq. (1).

For the fixed G, the optimal discriminator D* is –

$$D_G^* = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)} \quad (2)$$

When $p_{data}(x) = p_G(x)$, then Eq. (2) will be $1/2$. Putting this value in Eq. (1), we get the value of function $V(D, G) - \log 4$ which is also known as global minima [26].

Discriminator uses KL divergence and JS divergence to calculate how one image is different or similar from another image:

$$D_{KL}(P \quad Q) = \int p(x) \log \frac{p(x)}{q(x)} \quad (3)$$

$$JSD(P \quad Q) = \frac{1}{2} D_{KL}(P \quad M) + \frac{1}{2} D_{KL}(Q \quad M) \quad (4)$$

where $M = (p(x) + q(x))/2$.

KL divergence is not symmetric which can be understood by the following example [26]:

when $p(x) = 1$ and $q(x) = 0$, then $KL(P\|Q) = +\infty$ while $KL(P\|Q) = 0$ when $p(x) = 0$ and $q(x) = 1$ [18].

JS divergence is symmetric having value 1/2 log 2 in both cases [18].

Algorithm of Training GAN Training of GAN is difficult as we need to synchronize between two neural networks to train them. Following is the algorithm for training GAN using back propagation. The training of GAN is done in batches (Algorithm 1).

In the algorithm, k = no. of steps applies to the discriminator.

While training the GAN, we update the loss function of the discriminator by gradient ascent and update the loss function of the generator by gradient descent [4].

Problems in GAN While Training In the early stage of training, the generator does not generate appropriate data samples and the discriminator rejects the data sample with high confidence. As a result, the gradient of G saturates and does not get trained. This is known as the vanishing gradient problem. To avoid this problem, generators are trained to maximize $\log(D(G(z)))$ rather than minimize $\log(1 - D(G(z)))$ [4].

Another drawback of GAN is non-saturating game between discriminator and generator, that is, both discriminator and generator try to cancel out the values of each other to make zero sum value, that is, may be the current update leads one player downhill but same update can lead the other player uphill. In this procedure, there are oscillations toward the converging solution and both may not reach the converging point or may reach the converging point. This makes the GAN training unstable. It is studied that GAN converges to its minima when there is no mode collapse or parameters are updated within the function space. If this condition does not happen, GAN in most cases does not converge to its global minima [25].

The other big problem in GAN is mode collapse which is a big issue as it limits the GAN application. Mode collapse is a condition a model starts generating the same data or very less diverse data. Full mode collapse is occasional however partial mode collapse is seen more frequently. In a study, it is found that mode collapse can emerge when the max-min solution of GAN is not the same as min-max solution, that is, maximizing discriminator and minimizing generator is not the same as minimizing generator and maximizing discriminator:

$$G^* = \min_G \max_D V(G,D) \tag{5}$$

Algorithm 1 Algorithm of training GAN

	INPUT: n number of real images for discriminator and n number of noise data samples for generator.	
	OUTPUT: n number of images generated by generator.	
1	**for** number of training iterations do the following	
2		**for** k steps do the following
3		Take a sample of mini batch of n noise samples $\{z(1), \ldots, z(n)\}$ from noise prior $P_g(z)$.
4		Take a sample of a mini batch of n examples $\{x(1), \ldots, x(n)\}$ from data generating distribution $P_{\text{data}}(x)$.
5		Modify the discriminator by ascending its stochastic gradient value: $\nabla_{\theta_d} \frac{1}{n} \sum_{n}^{i=1} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]$
6		**end** for
7		Take a mini batch of n noise samples $\{z(1), \ldots, z(n)\}$ from noise prior $p_g(z)$.
8		Modify the generator by descending its stochastic gradient value: $\nabla_{\theta_g} \frac{1}{n} \sum_{n}^{i=1} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$
9	**end** for	

G∗ takes samples through the data distribution. When the order of min as well as max is changed

$$G^* = \max_D \min_G V\left(G,D\right) \tag{6}$$

find the minimum value with reference to the generator, at present it resides in the internal loop of optimization method. Then the generator mapped every z value to the corresponding x coordinate which the discriminator knows is mostly expected to be real in place of fake. Parallelly gradient descent does not distinctly prefer min-max over max-min and vice versa. It was used in the expectation that it will act like min-max however it frequently acts like a max-min equation. It is also found that mode collapse is not because of using JS divergence or KL divergence. Cost function also does not contribute to mode collapse. It may be the training of GAN which is causing mode collapse conditions [25].

What GAN Cannot Generate An experiment is done to analyze which modes are dropped by GAN while generating images. In the experiment of measuring as well as visualizing mode dropping in the most modern generative models, comparison is done using statistics analysis of the segmented images. Both the real and generated images are segmented and a deep analysis is done. Analysis is done at both levels – distributed as well as instance level using the images generated by WGAN, Progressive GAN, and Style GAN.

A visualization tool which predicts current analysis is developed by layer-wise inversion of the generator to see which mode is dropped by the tool. This tool is used for studies to get little perception of the phenomenon in modern GANs. According to the studies, it is found that GAN does not drop the modes or classes of objects because they are distorted or have low quality as in noise. Even they are not rendered as they are not part of the image. The dropped modes are hard to learn by GAN and that is why they are dropped, but this does not affect the quality of the image. It is still an open question as to which modes are dropped by GAN [11].

Methods and Algorithms to Reduce Mode Collapse A lot of methods and new GANs were introduced to reduce mode collapse from the GAN by the researchers after the introduction of GAN. Table 1 illustrates the list of new GANs and other methods which attempt to alleviate or reduce mode collapse since 2016.

These methods are specially designed to alleviate mode collapse from GAN and stable training. Unrolled GAN is the very first method which was introduced to reduce mode collapse in GAN.

Unrolled GAN works on the principle of calculating generator's gradient on maximum discriminator's value function. Ideally, the gradient for G is not calculated for the max value of D in $V(D, G)$ function and prone to mode collapse. To alleviate it, a graph structure is generated for the k steps of learning of the discriminator to maximize its value function and the gradient of the generator is calculated by back propagating through all k-step discriminators. It will take tens of thousands

of k steps to fully maximize the value function. But it is also experimented that 10 steps can also reduce mode collapse effectively. The value function for generator becomes:

$$f_K\left(\theta_G,\theta_D\right)=f\left(\theta_G,\theta_D^K\left(\theta_G,\theta_D\right)\right) \tag{7}$$

update value of generator and discriminator is as follows:

$$\theta_G \leftarrow \theta_G-\eta\frac{df_K\left(\theta_G,\theta_D\right)}{d\theta_G} \tag{8}$$

$$\theta_D \leftarrow \theta_D+\eta\frac{df\left(\theta_G,\theta_D\right)}{d\theta_D} \tag{9}$$

The disadvantage of this method is that it is computationally high and for large datasets, K may increase linearly, and this method is also not implemented for ImageNet. Further research is required to implement this method on ImageNet [12].

A lot of methods and techniques were introduced in the year 2017 and 2018. A description of these GANs are given below.

WGAN was introduced in 2017 with a concept of earth mover distance to calculate how one probability distribution differs from other probability distribution of the data. It uses earth-mover distance for divergence calculation rather than KL divergence or Jensen Shannon divergence or total variance which is given below:

$$W\left(\mathrm{P}_r,\mathrm{P}_g\right)=\inf_{\gamma\in\Pi\left(\mathrm{P}_r,\mathrm{P}_\theta\right)}\mathrm{P}_{(x,y)\sim\gamma}\left[\,x-y\,\right] \tag{10}$$

Earth mover distance guides how much mass needs to be transported to make both distributions similar.

It is found out that earth mover distance is weakest among them and KL is the strongest one to calculate the divergence between the two probability distributions. WGAN also modifies the objective function of GAN. Now in it, the discriminator tries to maximize $D(x) - D(G(z))$ and the generator tries to maximize $D(G(z))$. In

Table 1 Mode Collapse Reducing Methods of the GAN

S.no.	Year	Technique
1.	2016	Unrolled GAN
2.	2017	VAEGAN, VEEGAN, PacGAN, MAD-GAN, WGAN, distributional adversarial network, D2GAN, Bayesian-GAN
3.	2018	Primal dual subgradient GAN, MRGAN, MGGAN, sub-GAN, BourGAN
4.	2019	ProbGAN, MSGAN, spectral-regularization, diversity-sensitive GAN, MDGAN
5.	2020	Dropout-GAN, TailGAN
6.	2021	SSAT-GAN
7.	2022	UniGAN

WGAN, discriminator loss is also known as critic loss and its value is not only bound to between intervals [0, 1]. WGAN is proved to be very effective in mode collapse and also stables the training of GAN [27].

To resolve mode collapse, **VEEGAN** introduces a reconstructor network whose work is opposite to the generator, that is, to convert the generated data and true data into latent. This reconstructor helps the generator to guide which mode is dropped by comparing the latent of both true data and generated data. The mismatch between the latent leads to the conclusion that few modes are dropped. The generator and reconstructor network are trained jointly and for the training of the reconstructor network, the data with forgotten mode is ignored and this network uses an autoencoder with l2 loss functions. The objective function for VEEGAN is as follows:

$$O_{\text{entropy}}(\gamma,\theta) = E\left[\left\|z - F_\theta\left(G_\gamma(z)\right)\right\|_2^2\right] + H\left(Z, F_\theta(X)\right) \tag{11}$$

where $H(Z, F_\theta(X))$ is mapping between true data distribution $F_\theta(X)$ and gaussian distribution Z using cross entropy. VEEGAN is very much similar to ALI and BiGAN however its objective function provides remarkable benefits over the logistic regression loss. VEEGAN may be similar to other GANs also but it is different from them. Few of them are ALI, BiGAN, InfoGAN, etc. ALI and BiGAN both use the objective function same as vanilla GAN, so the output depends on the discriminator while in VEEGAN, the reconstructor term does not depend on the discriminator and can provide learning signals while the discriminator is constant and reduces mode collapse. InfoGAN and VEEGAN seem to be similar but the key difference between them is that InfoGAN does not train reconstruct network on the true data. In the experiment, it is found out that VEEGAN is very much suitable for identifying modes in comparison of ALI, BiGAN, Unrolled GAN, and vanilla GAN and produces much sharper images than others. It is also found that all GANs took almost the same time in computation except Unrolled GAN. VEEGAN runs suitable on default parameters so there is no hyper parameter tuning required [5].

The main idea of **PACGAN** is to modify the discriminator to construct decisions on the basis of several samples of the identical class which belong to either true data or generated data. The rest of the architecture is the same as the original GAN, having a generator and adversarial loss function. In PACGAN, the responsibility of discriminator is not just to map whether single data is real or fake, instead this augmented discriminator is used which maps m samples to a single class label. The intuition behind PACGAN is "Discriminator will detect mode collapse easily due to the reason that insufficiency of diversity is greater in groups rather than a single data." The discriminator is also known as "packed discriminator" and concatenated m samples are known as "degree of packing." The packing is implemented by increasing the number of nodes in the input layer of the discriminator by the factor m where m is the degree of packing. The rest of the architecture is the same. The advantage of PACGAN is that it is easy to implement and can be added to any GAN method. In the experiment, PacGAN is compared with the Unrolled GAN, VEEGAN, BiGAN, and ALI and found out that PacGAN captures more modes with

very less KL divergence. In stacked MNIST data, it captures around 1000 ± 0.00 modes with the KL divergence 0.07 ± 0.005 in generated images [7].

Dual discriminator GAN attempts to alleviate mode collapse from GAN by introducing one extra discriminator to the GAN architecture. In D2GAN one discriminator is rewarded for rejecting data generated by generator and the other discriminator favors that data by using KL divergence along with reverse KL divergence, and moreover the generator needs to satisfy both the discriminators. It is basically a three-player minimax game whose objective function is:

$$\begin{aligned}\min_{G}\max_{D_1,D_2}\mathcal{J}\left(G,D_1,D_2\right) &= \&\alpha\times \mathrm{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\log D_1\left(\mathbf{x}\right)\right]+\mathrm{E}_{\mathbf{z}\sim P_{\mathbf{z}}}\left[-D_1\left(G\left(\mathbf{z}\right)\right)\right]\\ &+\mathrm{E}_{\mathbf{x}\sim P_{\text{data}}}\left[-D_2\left(\mathbf{x}\right)\right]+\beta\times \mathrm{E}_{\mathbf{z}\sim P_{\mathbf{z}}}\left[\log D_2\left(G\left(\mathbf{z}\right)\right)\right]\end{aligned} \tag{12}$$

where α and β are newly introduced hyperparameters having value $0 < \alpha, \beta \leq 1$ to stabilize the learning and control the effect of KL and reverse KL. Experiment with MNIST data shows that D2GAN captures more modes than unrolled GAN, Reg GAN, DCGAN, and vanilla GAN [32].

GANs are implicit density models which learn to generate data using SGD. Due to some reasons it is not able to replicate a few modes and generate less diverse data. Bayesian GAN increases the diversity among generated data by including Bayesian inference to the model which comes under explicit density model. It uses prior distribution to calculate the posterior distribution. To marginalize the posteriors over the weights of discriminator and generator stochastic gradient Hamiltonian Monte Carlo method is used.

To infer posteriors over θ_g, θ_d, sampling is done by following conditional posteriors:

$$p\left(\theta_g \mid z,\theta_d\right)\propto\left(\prod_{n_g}^{i=1}D\left(G\left(z^{(i)};\theta_g\right);\theta_d\right)\right)p\left(\theta_g \mid \alpha_g\right) \tag{13}$$

$$p\left(\theta_d \mid z,X,\theta_g\right)\propto\prod_{n_d}^{i=1}D\left(x^{(i)};\theta_d\right)\times\prod_{n_g}^{i=1}\left(1-D\left(G\left(z^{(i)};\theta_g\right);\theta_d\right)\right)\times p\left(\theta_d \mid \alpha_d\right) \tag{14}$$

$p(\theta_g|\ \alpha_g)$ and $p(\theta_d|\ \alpha_d)$ are the priors over the parameters of generator as well as discriminator, along with the hyperparameters α_g and α_d, respectively. n_d and n_g are the numbers of the mini-batch samples for the discriminator as well as generator, respectively [31].

In 2017, not only these GAN methods are introduced, even a lot more GAN methods are introduced like VAEGAN which is a combination of variational autoencoder and GAN to overcome the disadvantages of both, that is, VAE generates blurry images and GANs are prone to mode collapse, both the models are combined together which become very effective in reducing mode collapse [33]. **MADGAN** include multiple generators to avoid mode collapse so that all generators capture every mode and there is less probability of a mode collapse. These generators are

combined together to get the overall results. These generators are called multi-agent diverse and the model is known as MADGAN [24]. A complex structure is the disadvantage of this model.

In 2018, a lot of GAN methods were introduced to mitigate mode collapse from GAN.

MGGAN (Manifold Guided GAN) tries to resolve the problem of mode collapse by introducing a guided network which inspires the generator to grasp all the modes of the data distribution rather than being stuck to a few modes. This guided network consists of pre-trained encoders and discriminators. The output of the discriminator in manifold space tells the generator how much latent of real data and fake data differ from each other and the generator updates its weight according to both of the discriminators' output. The objective function of MGGAN is:

$$\min_{D_x, D_m} \mathrm{E}_{x \sim P_{\text{data}}} \left[\log\left(D_x\left(x\right)\right) + \log D_m\left(E\left(x\right)\right) \right] + \mathrm{E}_{z \sim P_z} \left[\begin{array}{c} \log\left(1 - D_x\left(G\left(z\right)\right)\right) \\ + \log\left(1 - D_m\left(E\left(G\left(z\right)\right)\right)\right) \end{array} \right] \tag{15}$$

Generator's objective function:

$$\min_{G} - \mathrm{E}_{z \sim P_z} \left[\log\left(D_x\left(G\left(z\right)\right)\right) + \log\left(D_m\left(E\left(G\left(z\right)\right)\right)\right) \right] \tag{16}$$

Generator tries to meet the two goals – one is to find dissimilarity between real and fake data and another is to map real and fake data in manifold space. Manifold network is weakly connected to the rest of the network bidirectionally. Strict bidirectional mapping has the limitation. In strict bidirectional mapping, when additional constraints are applied on the encoder, the generator depletes the generation power to enclose the extensive range of the latent distribution and strict constraints rarely satisfied, which makes the network unstable and reduces representational power of the encoder that later results in a generation of low-quality images. So, in MGGAN, weak bidirectional mapping is used which is feasible and realistic and makes it to cover the missing modes and produce sharp images with diversity. MGGAN seems similar to ALI and BiGAN but in these GANs, the discriminator has two responsibilities – one to find out real and fake images and the other is to match joint probability distributions. This makes the discriminator insensitive to change for each distribution and not able to capture minor changes in the distribution and the whole network becomes unfaithful toward the images. MDGAN and VEEGAN use the reconstruction loss to reduce mode collapse but it is difficult to tune the balancing parameter in both as the network's loss unit and reconstruction loss unit is different. All these disadvantages are overcome in MGGAN. In the experiment, it is found that vanilla GAN is much prone to mode collapse. VEEGAN as well as Unrolled Gan capture all modes despite them generating very scattered data. It is observed that MGGAN captures every mode and reduces mode collapse [6].

Primal-dual subgradient GAN uses primal dual algorithm to avoid mode collapse from GAN. It is found that GAN does not face any mode collapse when some noise is added to it. Actually, it works as a regularizer for it. Inspired from this idea, primal-dual subgradient method is used where discriminator is primal variable and generator is dual variable. The dual variable is modified according to the subgradient of $L(x(t), \lambda(t))$ with reference to $\lambda(t)$ at every iteration t and primal variable is modified relative to the subgradient of $L(x(t), \lambda(t))$ with reference to $x(t)$ where L is a Lagrangian function [19].

Bour-GAN uses Bourgain's theorem to reduce mode collapse inspired from the idea that modes are geometric structures of the data distribution in the metric space. This experiment argues two basic questions:

The commonly used multivariate gaussian which generates random vectors for generator network.
Geometric interpretation of modes.

It was demonstrated in the experiment that in the presence of modes, selecting a random vector from a single gaussian leads to the larger gradient to the generator and different metrics may lead to different distributions of modes which cannot be interpreted. So, mixture gaussian and logarithmic pairwise distance distribution (LPDD) is used in BourGAN. In BourGAN, the very first step is to preprocess the dataset which includes subsampling to the dataset. In this process, m no. of data items is selected from the random distribution. This step is essential when the dataset is large as it reduces the computational cost. In the experiment it is found that the value of m = 4096 is sufficient for the dataset. After this, a gaussian mixture model is constructed to generate random vectors in latent space. In this process, data items are embedded to l_2 space. The model must be constructed in such a way that latent vector dimensionality must be small and mode structure must be reflected in the latent space. To achieve the goal, Bourgain embedding algorithm is used. After this, training is done. While training BourGAN, to alleviate mode collapse, the generator is encouraged to match the pairwise distance of generated samples to the pairwise distance of the l_2 latent vector in Z. The objective function of BourGAN is:

$$L(G,D) = L_{\text{gan}} + \beta L_{\text{dist}} \tag{17}$$

$$L_{\text{gan}}(G,D) = E_{x\sim X}\left[\log D(x)\right] + E_{z\sim Z}\left[\log\left(1 - D(G(z))\right)\right] \tag{18}$$

where L_{gan} is the standard GAN equation and β is a parameter to balance the two terms. In L_{dist}, z_i and z_j are the two separate samples where $z_i \neq z_j$. The advantages of using logarithm distance are:

(i) The outliers are prevented.
(ii) It turns the uniform scale of distance metrics into constant so that it has no effect on optimization.
(iii) It eases the theoretical analysis.

BourGAN is compared with standard GAN, VEEGAN, Unrolled GAN, and PacGAN and it captures more modes than others and prevents them from being generated again. Standard GAN is worst in capturing modes while other methods are close to BourGAN. Wasserstein distance is calculated to know how well the model is capturing data distribution. BourGAN performs the best among them [30].

MRGAN (Manifold regularized GAN) introduces a term manifold regularization to alleviate mode collapse and this idea can be extended on DCGAN and WGAN to improve the diversity of generated dataset. This model is inspired from the idea that real data lies in submanifolds and generated and real data which lies in disjoint manifolds is the reason for facing issues by GAN in training. The objective function of MRGAN is:

$$\min_{u\in U}\max_{v\in V} E_{x\sim D_{\text{real}}} E_{y\sim D_{G_u}} \begin{bmatrix} \phi\left(D_v\left(x\right)\right)+\phi\left(1-D_v\left(y\right)\right) \\ ++\lambda \int_{x\sim M} \nabla_M\left(\psi\left(y\right)-\psi\left(x\right)\right)^2 dPx \end{bmatrix} \tag{19}$$

where $\lambda\int_{x\sim M} \| \nabla_M(\psi(y) - \psi(x))\|^2$ is a regularizing term. It is theoretically proved that the MRGAN provides stable training with very less matching score between real data and generated data (shown in experiment) and forced the generator to generate a unique dataset [20].

In 2019, most appropriate and compact methods were introduced to resolve the problem of mode collapse and most of the techniques were different types of regularization. Few of them were specially designed for a particular type of GAN and few were general.

SRGAN (spectral regularization GAN) works on the idea that mode collapse occurs because of spectral collapse in GAN. In this experiment, it is found that there is no mode collapse seen when singular values in $\bar{W}_{SN}\left(W\right)$ in the discriminator are very close to 1 and when the singular value drops dramatically, mode collapse occurs. This phenomenon of dropping a large no. of singular values is known as spectral collapse where $\bar{W}_{SN}\left(W\right)$ is spectral normalized weight matrix. When spectral collapse occurs, there is a high probability of occurring mode collapse and Inception score as well as Fréchet inception distance also drops. So, to alleviate mode collapse, spectral collapse must be removed. SRGAN is used to do so. A regularization is done to remove spectral collapse. In SRGAN, singular value decomposition is used. So, the weight matrix can be expressed as follows:

$$W = U\cdot\Sigma\cdot V^T \tag{20}$$

where U and V are orthogonal matrix, U is left singular matrix of W while V is right singular matrix of W and Σ is:

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \tag{21}$$

where $D = diag\ \{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ represents the spectral distribution of W.

When mode collapses, spectral distribution condenses on the first singular value and rest values drop. To balance out this, ΔD is applied where ΔD is $diag\{\sigma_1 - \sigma_1, \sigma_1 - \sigma_2, \ldots, \sigma_1 - \sigma_i, 0, \ldots, 0\}$, and i is a hyperparameter ($1 \leq i \leq r$). The suitable value of i is 0.5 N (proved in the experiment) and used in SRGAN. This experiment demonstrated that SRGAN performed very well over SNGAN (Spectral normalized GAN) and treated SNGAN as a special case in SRGAN [23].

Diversity sensitive GAN is a regularization technique to reduce mode collapse from conditional GAN. The intuition behind DSGAN is that mode collapse occurs when the generator maps the large portion of latent codes to the same output, that is, $G(x, z_1) \approx G(x, z_2)$ for all $z_1, z_2 \sim N(0, 1)$. To avoid this, regularization is done to the generator and the regularization function is:

$$\max_G L_z(G) = E_{z_1, z_2}\left[\min\left(\frac{G(x,z_1) - G(x,z_2)}{z_1 - z_2}, \tau\right)\right] \tag{22}$$

where τ bound is for numerical stability. Its objective function is [9]:

$$\min_G \max_D L_{cGAN}(G,D) - \lambda L_z(G) \tag{23}$$

where λ controls the degree of stochasticity in generator.

This regularization technique is simple, general, and can be used with other existing conditional GAN's objective. This regularization also provides control over diversity via hyperparameter λ. It can also be extended to incorporate different distance metrics to measure the diversity. This is shown using distance in feature space and for sequence data in the experiment. Three types of tasks are performed using DSGAN, which are image to image translation, video prediction, and image inpainting. It is found that DSGAN performs very well and is easy to integrate with the conditional GAN and generate diverse dataset [9].

MSGAN (Mode seeking GAN) is also a regularization technique introduced for conditional GANs as they need to generate diverse images each time for the same input. It works on the idea that mode collapse between two images mostly occurs when the latent space of these two images is very close to each other in the latent space. The distance between them is maximized during the regularization. The objective function of MSGAN is:

$$L_{\text{new}} = L_{ori} + \lambda_{ms} L_{ms} \tag{24}$$

where L_{ori} is the original objective function and L_{ms} is

$$L_{ms} = \max_G\left(\frac{d_I(G(c,z_1), G(c,z_2))}{d_z(z_1,z_2)}\right) \tag{25}$$

Mode seeking GAN is experimented on conditional GAN, image to image translation as well as text to image synthesis, therefore as a result it is found that it performs very well both in terms of visual quality and diversity and provides baseline for the GAN. For the evaluation purpose, some metrices are used which are Fréchet Inception Distance (FID), Learned Perceptual Image Patch Similarity (LPIPS), and Number of Statistically-Different Bins (NDB) and Jensen-Shannon Divergence (JSD). FID is used to test the quality of images. LPIPS is used for diversity measurement for image and JSD and NDB are used to measure the similarity between original and generated images. This GAN method is used on three conditional GANs which are image-to-image translation, categorical generation, as well as text-to-image synthesis and found to be an effective method and generated diverse data [22].

Mixture density GAN (MDGAN) reduces mode collapse by introducing the concept of mixture density of images. It uses the mixture Gaussians in the objective function of which the mean vectors are put down in vertices of the d dimensional simplex. In MDGAN, discriminator creates clusters on the basis of embeddings of real images. Generator tries to generate images which are very close to the embedding of the real images to fool the discriminator. As there are multiple clusters, every mode of images is captured and alleviates mode collapse.

The objective function of MDGAN is given below:

$$\min_G \max_D \mathcal{L}(G,D) = \min_G \max_D (\mathrm{E}_{x\sim p_{\text{data}}}\left[\log\left(lk\left(D(x)\right)\right)\right] + \mathrm{E}_{z\sim p_z}\left[\log\left(\lambda - lk\left(D\left(G(z)\right)\right)\right)\right]) \tag{26}$$

where the $lk(e)$ is also known as the likelihood for given image, let $e = D(x)$ which is defined in the given Eq. (27):

$$lk(e) = \sum_{C}^{j=1} \frac{1}{d+1} \cdot \Phi\left(e;,\mu_j;,\Sigma_j\right) \tag{27}$$

where the following terms stands for:

Φ = Gaussian PDF.
μ_j=mean vector
$\sum_j$ =covariance matrix for Gaussian component j
C = total number of Gaussian components.

(Each mixture weight equals 1/1 + d)

The discriminator firstly encodes the inserted image x within embedding e, when it tries to differentiate between real and fake image. Then, it calculates likelihood $lk(e)$. The $lk(e)$ is considered as the probability of e, like an embedding of the real image to the given present model. In the experiment, MDGAN is compared with the other seven baseline GANs which are Vanilla GAN, ALI, Unrolled GAN, VEEGAN, DeliGAN, InfoGAN, SpecNorm GAN, then it is found that MDGAN performs

better than other GANs and captures each mode. MDGAN has more FID score than other GANs [34].

ProbGAN generally works on the idea of the aggregation of the generators to single discriminator. It is believed that a single generator cannot capture all the modes of the data. To capture the multi-modal data distribution, there is a requirement of multiple generators. In ProbGAN, multiple generators are used with probabilistic network. It seems to be similar to Bayesian GAN but it is different from it. In the studies, it is found that Bayesian GAN does not converge due to the use of weak informative prior but ProbGAN overcomes this disadvantage. ProbGAN uses less standard prior. It sets different priors for generator and discriminator. For the generator, previous time step prior is used. The intuition behind this is that when the generator generates plausible data which is close to true data the discriminator tries to give equal score which results in equal likelihood. So, it is better to keep the differences between generator and discriminator by setting previous time step prior to the generator. This dynamically evolving prior results in the generation of diverse data and converges to some point. For the posterior calculation, Stochastic Gradient Hamiltonian Monte Carlo method is used. In the experiment, ProbGAN is compared with DCGAN, Mixture GAN, and Bayesian GAN on the dataset STL-10, CIFAR-10, as well as image-net over the evaluation metric Inception score along with FID. And it is found that ProbGAN has highest inception score and lowest FID score [10].

Dropout is a regularizing technique which is utilized to avoid the problem of overfitting in neural networks by dropping some neurons from the network randomly based on probability. Dropout-GAN uses this technique to drop the discriminator from the set of K discriminators as it works on intuition that mode collapse occurs due to overfitting of a model over a particular mode due to satisfying conditions on a single discriminator. It was introduced in 2020.

The dropout-GAN generator updates its value according to every discriminator which is not dropout. And every discriminator also updates its weight according to the GAN's standard equation. There may be a condition that every discriminator is dropped out and no one is left to guide the generator. In this case, a discriminator is randomly chosen among the set of discriminators and generator's weights are updated. The function used for this neural network along with standard GAN equation is: given below:

$$\min_{G}\max_{\{D_k\}}\sum_{K}^{k=1}V\left(D_k,G\right)=\sum_{K}^{k=1}\delta_k\left(\mathrm{E}_{x\sim p_r(x)}\left[\log D_k\left(x\right)\right]+\mathrm{E}_{z\sim p_z(z)}\left[\log\left(1-D_k\left(G\left(z\right)\right)\right)\right]\right) \tag{28}$$

Dropout GAN uses dynamic ensemble discriminators for training keeping the original GAN objective function. In dropout GAN, each discriminator is trained individually. That means no other discriminator knows other discriminator's existence since changes were not made on their individual gradient updates. In this GAN method, generator is updated by a drastically large value as it needs to satisfy

multiple discriminators. As a result, it is found that it converges to zero very easily. It is also noticed in experiment that as the number of discriminator increases, the number of generator's gradient starting to converge also increases. It means that a large number of discriminators can delay the learning process. In the experiment, batch partitioning is done among the different discriminators so that every discriminator learns a specific mode of the data distribution. In the experiment, it is also found out that GAN works very well with dropout rate 0.2 and 0.5. Dropout GAN is compared with other GANs and it is found that dropout GAN performs well among them. Fréchet distance is calculated for the comparison and Dropout GAN performs well with less Fréchet distance and produces more diverse data than LSGAN, MODGAN, DRAGAN [8].

TailGAN is a type of GAN in which we can perform AD (anomaly detection) as well as generating samples at the back end of a typical distribution sample. The TailGAN utilizes adversarial training as well as GANs. After that it reduces the objective cost function by using the following equations:

$$L_{tot}\left(\theta,\mathbf{z},\mathbf{x},G\right) = \&w_{pr}L_{pr}\left(\theta,\mathbf{z},G\right) + w_{d}L_{d}\left(\theta,\mathbf{z},\mathbf{x}\right) + w_{e}L_{e}\left(\theta,\mathbf{z},G\right) + w_{sc}L_{sc}\left(\theta,\mathbf{z}\right) \tag{29}$$

where the following terms stand for:
L_{tot}=total cost function
L_{pr}=probability cost
L_{d}=distance loss
L_{e}=maximum-entropy cost
L_{sc} =scattering loss

$$\begin{aligned} L_{tot} = \frac{1}{N}\sum_{N}^{i=1}[&w_{pr}p_{g}\left(T\left(\mathbf{z}_{i};\theta\right)\right) + w_{d}\min_{j=1}^{M}T\left(\mathbf{z}_{i};\theta\right) - \mathbf{x}_{jp} \\ &+w_{e}p_{g}\left(T\left(\mathbf{z}_{i};\theta\right)\right)\log\left(p_{g}\left(T\left(\mathbf{z}_{i};\theta\right)\right)\right) \\ &+w_{sc}\frac{1}{N-1}\sum_{N}^{j=1,j\neq i}\frac{\mathbf{z}_{i}-\mathbf{z}_{jp}^{q}}{T\left(\mathbf{z}_{i};\theta\right)-T\left(\mathbf{z}_{j};\theta\right)_{p}^{q}}] \end{aligned} \tag{30}$$

By making use of the GAN that can straightforwardly calculate the probability density, we carry out sample generation at the back end of data distribution, it addresses distributions of multimodal alongside disconnected components and after that it also reduces mode collapse. We construct a tail formation model based out of GAN (TailGAN) for detection of any inconsistency. This type of GAN was created for sample generation at the back end of the data distribution as well as detecting any anomaly nearby the supporting boundary. With this TailGAN, we make use of the maximum amount possible of entropy regularization. By using this type of GAN that is itself learning the probability of the underlying distribution has many benefits of refining anomaly detection by authorizing to design a generator for generating boundary samples. The TailGAN addresses support alongside disjoint components as well as attains better performance in the images [3].

Mode Collapse Reducing by Making Use of UniGAN (Uniform Generator) In this method the UniGAN is used which is a generative framework alongside a generator which is based out of the normalizing flow. It is a simple GAN but it is yet sample efficient to generate uniformity regularization which in turn can be easily adapted to any other type of generative framework. Udiv which is a new type of diversity metric is also suggested along with this type of GAN for the assessment of uniform diversity under the set of generative samples given. The regularization for consistency of the generator executed is formed by the following:

$$\mathcal{L}_{\text{gunif}} \triangleq \mathrm{E}_{z\sim p_z}\left\{\left(\mathrm{LT}_g(z)-a\right)^2+\left(\frac{1}{\mathrm{LT}_{g^\dagger}(g(z))}-a\right)^2\right\} \tag{31}$$

After that we will put together the above defined NF-based generator as well as the normal discriminator within the generative framework giving out the GAN whose objective function is defined by the following equation:

$$\mathcal{L}_{\text{unigan}} = \mathcal{L}_{\text{gan}} + \lambda_{\text{gunif}}\mathcal{L}_{\text{gunif}} \tag{32}$$

where $\mathcal{L}_{\text{gan}}$ stands for original objective of GAN and balancing hyper-parameter is given by λ_{gunif}.

It was also proposed for maximizing the uniform diversity. By carefully examining the results experimented, it proves the effectiveness of the UniGAN framework to reduce the mode collapse [1].

SSAT-GAN (Spectral–Spatial Attention GAN) is a semi-supervised feature extraction technique that adds raw data into the framework of deep learning. Firstly, the unlabeled data is included in the discriminator for reducing the complication of samples training as well as by using the adversarial training it imparts an actual HSI (Hyperspectral Image) data distribution which is reconstructed. SSAT was also built to enhance the representation of HSIs and then it is extended to the discriminator along with the generator for extracting selective characteristics through ample spatial contexts as well as spectral signatures. Also, the SSAT modules grasp the 3D filter bank alongside SSAT weights for obtaining relevant feature maps for improving the distinct feature of the feature representation. For reducing mode collapse of GAN in unsupervised learning, the mean minimization loss is implemented. The loss function L_G formed in this type of GAN is given by:

$$\begin{aligned} L_G(q_D,q_G) &= -E_{z\sim p_z}\log\left(1-D\left(G(z;q_D)\right)[0]\right) \\ &= -E_{z\sim p_x}\log\left(1-D(Z;q_D)[0]\right) \\ &= -E_{z\sim p_z}\log\left(1-\bar{Y}^1[0]\right) \end{aligned} \tag{33}$$

The SSAT-GAN enhances the spreading of the characteristics with the extended SSAT feature for representing the feature. For improving characterization during the phase of feature extraction, this GAN effectively put on the attention weights for intensifying spectral bands as well as spatial correlations. For extracting discriminative features, SSAT-GAN put together the spectral as well as spatial attention modules in each of the discriminators together with generator composing of convolutional and transposed layers, respectively [2].

To solve the problem of mode collapse we had studied certain types of GANs in which the mode collapse is reduced either by introducing a new model in the GAN method or regularizing the latent or by adding some additional terms to the GAN loss function. Few of the methods discussed till now are helpful to solve the problem of mode collapse for only specific types of GAN. This shows that the problem of mode collapse is not solved globally or if it is solved, then either it involves very high computation or has very complex design or introduces some new hyperparameter to the loss function which in turn needs to be tuned further for good results.

3 Conclusion

GAN has become a very interesting and exciting topic nowadays and has a lot of applications. The study of GAN opens doors of lots of opportunity. In this chapter we discuss about the generative models, their classification on the basis of maximum likelihood. Then we deeply discuss the generative adversarial network along with the adversarial idea and its training algorithm. There is small talk about the GAN's training issues and understanding what is mode collapse. A thought is discussed about what GAN cannot generate and then there is a long discussion on the different types of GAN methods which try to alleviate mode collapse from GAN along with their objective function.

References

1. Pan, Z., Niu, L., & Zhang, L. (2022). UniGAN: Reducing mode collapse in GANs using a uniform generator. In *Advances in neural information processing systems*.
2. Liang, H., Bao, W., Shen, X., & Zhang, X. (2021). Spectral–spatial attention feature extraction for hyperspectral image classification based on generative adversarial network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14*, 10017–10032.
3. Dionelis, N., Yaghoobi, M., & Tsaftaris, S. A. (2020). Tail of distribution GAN (TailGAN): Generative adversarial-network-based boundary formation. In *2020 sensor signal processing for defence conference (SSPD)* (pp. 1–5). IEEE.
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems, 27*.

5. Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., & Sutton, C. (2017). Veegan: Reducing mode collapse in GANs using implicit variational learning. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 3310–3320).
6. Bang, D., & Shim, H. (2018). Mggan: Solving mode collapse using manifold guided training. *arXiv preprint arXiv*, 1804.04391.
7. Lin, Z., Khetan, A., Fanti, G., & Oh, S. (2018). Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems.*
8. Mordido, G., Yang, H., & Meinel, C. (2018). Dropout-GAN: Learning from a dynamic ensemble of discriminators. *arXiv preprint arXiv*, 1807.11346.
9. Yang, D., Hong, S., Jang, Y., Zhao, T., & Lee, H. (2019). Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv*, 1901.09024.
10. He, H., Wang, H., Lee, G. H., & Tian, Y. (2018). Probgan: Towards probabilistic GAN with theoretical guarantees. In *International conference on learning representations*.
11. Bau, D., Zhu, J. Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., & Torralba, A. (2019). Seeing what a GAN cannot generate. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4502–4511).
12. Metz, L., Poole, B., Pfau, D., & Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. *arXiv preprint arXiv*, 1611.02163.
13. Yang, S., Xie, L., Chen, X., Lou, X., Zhu, X., Huang, D., & Li, H. (2017). Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 685–691). IEEE.
14. Wang, T., Zhang, T., & Lovell, B. (2021). Faces à la carte: Text-to-face generation via attribute disentanglement. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3380–3388).
15. Mao, W., Lou, B., & Yuan, J. (2019). TunaGAN: Interpretable GAN for Smart Editing. *arXiv preprint arXiv*, 1908.06163.
16. Park, S. J., Son, H., Cho, S., Hong, K. S., & Lee, S. (2018). Srfeat: Single image super-resolution with feature discrimination. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 439–455).
17. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In *International conference on machine learning* (pp. 7354–7363). PMLR.
18. Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2020). A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv*, 2001.06937.
19. Chen, X., Wang, J., & Ge, H. (2018). Training generative adversarial networks via primal-dual subgradient methods: A Lagrangian perspective on GAN. *arXiv preprint arXiv*, 1802.01765.
20. Li, Q., Kailkhura, B., Anirudh, R., Zhou, Y., Liang, Y., & Varshney, P. (2018). MR-GAN: Manifold regularized generative adversarial networks. *arXiv preprint arXiv*, 1811.10427.
21. De Cao, N., & Kipf, T. (2018)s. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv*, 1805.11973.
22. Mao, Q., Lee, H. Y., Tseng, H. Y., Ma, S., & Yang, M. H. (2019). Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1429–1437).
23. Liu, K., Tang, W., Zhou, F., & Qiu, G. (2019). Spectral regularization for combating mode collapse in GANs. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6382–6390).
24. Ghosh, A., Kulharia, V., Namboodiri, V. P., Torr, P. H., & Dokania, P. K. (2018). Multi-agent diverse generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8513–8521).
25. Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv*, 1701.00160.
26. Uddin, S. N. (2019). GAN-Intuitive Approach to Mathematics.

27. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223). PMLR.
28. Hartmann, K. G., Schirrmeister, R. T., & Ball, T. (2018). EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals. *arXiv preprint arXiv*, 1806.01875.
29. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., & Malossi, C. (2018). Bagan: Data augmentation with balancing GAN. *arXiv preprint arXiv*, 1803.09655.
30. Xiao, C., Zhong, P., & Zheng, C. (2018). Bourgan: Generative networks with metric embeddings. *arXiv preprint arXiv*, 1805.07674.
31. Yunus, S., & Wilson Andrew, G. (2017). Bayesian GAN. In *Proc. of the conference on advances in neural information processing systems (NeurIPS)* (pp. 3622–3631).
32. Nguyen, T. D., Le, T., Vu, H., & Phung, D. (2017). Dual discriminator generative adversarial nets. *arXiv preprint arXiv*, 1709.03831.
33. Rosca, M., Lakshminarayanan, B., Warde-Farley, D., & Mohamed, S. (2017). Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv*, 1706.04987.
34. Eghbal-zadeh, H., Zellinger, W., & Widmer, G. (2019). Mixture density generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5820–5829).

Medical Image Synthesis Using Generative Adversarial Networks

Vishal Raner, Amit Joshi, and Suraj Sawant

1 Introduction

The development of deep neural networks and their cutting-edge performance in various imaging-related tasks has contributed to the arrival of third artificial intelligence boom and major changes in image processing. Most of the practitioners make use of computer-aided diagnosis because certain diseases cannot be diagnosed manually with bare eyes, so in order to diagnose such diseases many researchers proposed supervised as well as unsupervised learning algorithms in respective imaging domains [1]. The diagnosis of medical imaging systems such as retinal imaging, pathology, radiology, and dermatology mostly rely on supervised learning algorithms [2]. The training process of supervised algorithms requires annotated datasets which contains images having ground truth class or label associated with them. The annotated medical imaging related datasets are not easily available due to legal restriction and patient's privacy. Also it is very costly to produce a dataset; it is a tedious and time-consuming process to perform manual annotation of these acquired images in order to use it in supervised image analysis tasks. Because there is currently limited availability to large annotated datasets, deep learning approaches in medical diagnosis are still restricted. Deep learning algorithms, however, struggle with over-fitting when trained on small datasets, which is especially problematic when working with medical images. The varying incidences of different diseases frequently results in imbalanced datasets, particularly in medical images this issue further complicates the learning process. Therefore, it is necessary to find alternative ways to generate high-fidelity medical images for training supervised medical

V. Raner (✉) · A. Joshi · S. Sawant
Department of Computer Engineering and IT COEP Technological University (COEP Tech), Pune, Maharashtra, India
e-mail: ranervv21.comp@coeptech.ac.in; adj.comp@coeptech.ac.in; sts.comp@coeptech.ac.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_3

diagnostic systems. This work proposed a variant of Generative Adversarial Network (GAN), a Deep Convolution GAN (DCGAN) to synthesize retinal fundus images. Creating pictures of the retinal vasculature with a more detailed retinal vein network is the goal of this literature.

Discriminator and generator are the two network models that make up the GAN. The generator with the help of random noise vector generates fake images. The produced image is then fed to discriminator module; the task of it is to classify it as fake or real. The generator model weights get updated each time discriminator identifies generated image as fake. Both models get trained simultaneously. The generator model tries to learn the probability distribution function of underlying training set data in order to generate new examples to fool the generator. The generators do not start generating images that are similar to training data in pixel values; rather it starts randomly generating images. As training proceeds generator starts capturing underlying patterns in training data distribution and generates images which are similar to training data. Both generator and discriminator work on single objective function; like game theory generator tries to maximize it on the other hand discriminator tries to minimize it by producing plausible images which can fool discriminator.

Generative modeling and image synthesis reached previously unheard-of quality levels after Goodfellow et al. introduced Generative Adversarial Networks (GANs) [3]. Each iteration of the research on GANs pushed the boundaries of image quality farther and further. With growing strength, the GANs are creating magnificent photorealistic pictures that copy the information in the datasets they have been learned to replicate. GANs can generate data without defining the probabilistic model, which makes them data-generating tools. However GANs are also proven effective in various computer vision related tasks in the medical imaging domain [4]. GANs are an exciting and rapidly changing field that offer potential for generative models, as they can generate realistic examples of various problem domains [5]. It is possible to create a high-resolution RGB image such that even experts fail to identify between real used for training and synthetic image generated by the generator of GAN. Frechet Inception Distance (FID) score [6] was calculated using training data obtained from well-known and regularly used datasets to evaluate the clarity of the pictures produced using GANs.

1.1 Retinal Image Analysis

We can learn about the state of the entire system and the health of the eyes through retinal vessel network analysis. Ophthalmologists can identify early indications of vascular burden brought on by diabetes complications and hypertension and also the lethal retinal diseases like Retinal Artery Occlusion (RAO) and Retinal Vein Occlusion (RVO), which are brought on by abnormalities in vascular structure. Medical images, in particular, are inadequate, costly, and have restricted uses due to legal concerns such as patient's privacy. Furthermore, the size and annotation of the

datasets of medical images that are made publicly available are frequently inconsistent. Because of this, they are less effective for training data-hungry neural networks. The medical diagnosis mostly relies on supervised learning algorithms which require images along with their ground truths [7]. Due to these limitations it is hard to train such supervised diagnosis systems. This has a direct impact on how quickly medical diagnosis systems can develop.

In applications where neural networks are being used to generate images, it requires up-sampling from low resolution to higher resolution. The up-sampling task in convolutional layers is carried out by one of the interpolation methods. It is similar to manual feature engineering, that is, interpolation has to be selected while deciding network architecture, hence network cannot learn anything from it. After applying convolutional operation, positional connectivity gets established between input and output in forward direction, that is, top right values in input matrix affects top right values in output matrix.

This work proposed a DCGAN-based approach where the generator only consists of transposed convolutional layers as hidden layers. Transposed convolutional layers form the same connections as regular convolutions, but in the opposite direction (backward direction). Transposed convolution is used to carry out up-sampling, hence no requirement of fixed interpolation methods. The weights in transposed layers can be learned via backward connectivity, which helps the network up-sample data in the best way possible.

The remainder of the paper is structured as follows. In Sect. 2, we describe the Generative Adversarial Network and the previous work in this field. Section 3 elaborates about the methodology used. The experimental design and findings were covered in Sect. 4. In Sect. 5 elaborates about conclusions and findings of the proposed method.

2 Related Work

Wang et al. gave a detailed overview of deep learning based methods for medical image synthesis along with their clinical applications. Authors mentioned the need of additional quality assurance methods to identify abnormal generated images which are not compatible with imaging protocols [8]. Krithika et al. discussed various GAN architectures and their applications in medical imaging domain. Authors have enlisted various use cases of GANs in the medical domain such as image synthesis, segmentation, image reconstruction, registration, and classification [9]. Koshino et al. discussed about usefulness of GANs in medical imaging with respect to seven topics, namely, (I) data augmentation, (II) modality conversion, (III) denoising, (IV) image reconstruction for, (V) super-resolution, (VI) domain adaptation knowledge, and (VII) image generation with radio genomics and disease severity [10]. Yi et al. gave a detailed summary on medical image reconstruction publications, here authors enlisted dataset used, GAN methods used, qualitative measures, and loss functions used. Authors also gave a detailed summary of loss

Table 1 Batch-wise FID score of generated images

Batch number	FID score
B-1	150.8387
B-10	110.8625
B-20	88.1051
B-30	84.0791
B-40	79.74568
B-50	57.3344
B-60	48.4354
B-70	49.5234
B-70	49.5234

functions used for GAN implementations and quantitative measure [11]. Skandarani et al. tested different GAN architectures on three medical image modalities, namely, RGB retina, cardiac cine-MRI, and liver CT images. Authors used FID score to measure the performance of GAN-generated images in terms of visual acuity. Authors further trained U-Net on the produced images and checked the segmentation accuracy with original data. Authors found out that images generated can fool experts visually, but segmentation results reflected that GANs are far away in capturing the full richness of medical data [12]. Bissoto et al. used the variety of GAN architectures to generate and enhance skin lesion pictures. Authors further used anonymization techniques where real data gets replaced by synthetic data and the model performance is checked. Authors used FID score for the qualitative measure of generated images [13] (Table 1).

Andreini et al. developed a deep learning and GAN-based method for semantic segmentation of colonies of bacteria seen in photos of agar plates. Authors used CNN to separate bacterial colony from backgrounds, then used GAN to generate synthetic samples of these separated colonies. Authors then imposed these synthetic data on existing backgrounds of agar plate. Further, authors used both the real and synthetic data to train the network and achieved performance gain [14]. Chlap et al. gave a detailed overview of data augmentation techniques. Authors discussed about different GAN networks used for various image modalities. Authors also discussed about deformable techniques which allow complex form of data augmentation [15]. Ghassemi et al. proposed a new deep learning method to perform tumor classification task in MRI images. Initially authors pre-trained the network as discriminator in GAN on various MRI image datasets. Authors then placed fully connected layers and full network is trained as a classifier in order to distinguish among three classes of tumor. Authors achieved significant performance gain than other state-of-the-art methods [16]. Zhu et al. discussed about limitations of use GAN trained on natural images for medical image synthesis. Authors proposed lesion focused single image super resolution (LF-SR) technique for brain tumor MRI images to overcome this issue. Authors tested various GAN architectures, namely, WGAN, WGAN-GP, MS-GAN; they found that MS-GAN along with LFSR performed better [17].

Ma et al. proposed a residual neural network (ResNet) and DC-GAN based blood cell image classification framework. The authors implemented a novel loss function

in this paper and evaluated the framework on White Blood Cell (WBC) pictures, achieving a classification accuracy of 91.7% [18]. Joshi et al. proposed a GAN-based technique to improve MRI image resolution. Authors have trained SRGAN to convert low resolution blurry images to more detailed high resolution by 4x upscaling factor. Authors used peak signal to noise ratio (PSNR) and structural similarity index (SSIM) as performance metric and achieved a score of 6.403804 and 0.787986, respectively [19]. Nema et al. proposed the (RescueNet) residual cyclic unpaired encoder-decoder network which uses an unpaired adversarial learning method which segments whole tumor and performance enhancement in regions in brain MRI scan. Authors have used DICE and sensitivity measure as performance parameters and tested results on BraTS 2017 and BraTS 2015 dataset [20]. Rashid et al. proposed GAN-based data augmentation and skin lesion classification technique. Authors first generated synthetic images, which then fed to discriminator to predict the class of it. Authors achieved the balanced accuracy score of 0.861 [21]. Qin et al. proposed Style-GAN based skin lesion image synthesis technique. Authors have constructed the classifier on pre-trained deep Neural-Network with the help of transfer learning. Authors used the FID and Inception score (IS) as performance metrics. Authors achieved improvement of balanced multiclass accuracy, sensitivity, accuracy, specificity, and average precision by 5.6%, 24.4%, 1.6%, 3.6%, and 23.2%, respectively, than the CNN-based model [22]. Nie et al. implemented a completely convolution-based network model for the synthesis of medical images. Authors also applied Auto-context model for context aware image synthesis. The proposed model is tested on CT, 3 T MRI, and 7 T MRI. Authors addressed the synthesis of CT from MRI image and 7 T MRI from 3 T MRI image. Authors have used MAE and PSNR as performance measure [23]. Zhou et al. suggested (HI-Net) model to synthesize MRI images from several modalities. Authors have exploited the correlations present among multiple image modalities by means of the layer-wise multi-modal fusion strategy. Authors have used PSNR, SSIM, and Normalized Mean Square Error (NMSE) as a performance measure [24].

Sun et al. proposed an ANT-GAN image synthesis model to synthesize normal medical images based on their corresponding abnormal counterpart. The author tested the model on (BratS18) and liver tumor segmentation challenge dataset (LiTS). Authors used PSNR and VERISIMILITUDE score (VS) as performance metrics [25]. Devi et al. proposed DR-DCGAN to synthesize diabetic retinopathy images. Authors used resnet50 model to check the classification accuracy using synthesized images. Authors used APTOS Blindness dataset, and achieved classification accuracy of 98.7% [26]. You et al. gave n detailed survey of various applications of GAN-based techniques in ophthalmology domain. In this literature authors have discussed about various types of GANs used, imaging domains, and corresponding results obtained [27]. Iqbal et al. proposed MI-GAN model for retinal image synthesis. The proposed model can synthesize retinal images along with their segment mask. Authors tested the proposed model on STARE and DRIVE dataset. Authors used DICE coefficient score as performance metric and achieved the values 0.837 and 0.832 for STARE and DRIVE dataset, respectively [28]. Shenkut et al. proposed a two-stage GAN-based technique to synthesize fundus images. At first

stage the segmented vessel tree is extracted, the second stage performs image to image translation and produces fundus image. Furthermore, authors have built a classification model which achieved 0.872 validation accuracy [29]. Liang et al. proposed an C-GAN method for retinal image synthesis. Authors have used VGG-19 feature extraction module in generator part. Based on combined loss, the generator model is constrained to generate visually high-quality images only. Authors used Frechet Inception Distance (FID) and Sliced Wasserstein Distance (SWD) as performance metrics [30]. Yu et al. proposed a multi-channel multi-landmark (MCML) GAN-based image synthesis technique to synthesize retinal images. Authors used DRIVE and DRISHTI-GS dataset to train pix2pix-GAN and cycle-GAN architectures. PSNR and SSIM metrics are used as performance measures [31]. Beers et al. proposed a PGGAN-based image synthesis model for retinal and MRI image synthesis. Authors used Retinopathy of Prematurity (ROP) and multi-modal MRI images of glioma as dataset to train the proposed model. The authors applied the cutting-edge segmentation method and predicted segmentation masks created by algorithms utilizing GANs. The authors obtained an AUC score of 0.97 [32].

3 Proposed Methodology

In this work we have implemented DCGAN to synthesize retinal vasculature images. Convolution layers are used in place of fully connected GAN layers in the DCGAN model. The overview of the entire architecture is shown in Fig. 1. In the GAN architecture the generator is a forger attempting to generate data that appears to be real. Although it does not know what the actual data is, it can adapt thanks to the discriminator model's feedback. The discriminator compares the generated data with actual data, in our case generated images with real images and tries to identify it as real or fake. The back-propagation of the generator model is assisted with the help of output of the discriminator network. Conceptually, the GAN may be thought of as the generator and discriminator models competing in a min-max game. Both the models are trained simultaneously where the discriminator model tries to minimize the loss while the generator tries to maximize the loss. The generator network receives the random noise z as an input, using z to generate the output picture G (z). The discriminator accepts a synthesized picture (G(z)) or a real image (X) as input and outputs D, where D has a value between [0, 1].

The equation of loss function of GAN can be summarized as shown below in eq. 1 based on original paper by Ian Goodfellow. G (z) is the generator's output image produced using random noise z. D(x) is the discriminators output probability that the original image x is real. D (G (z)) is the output of discriminator indicating the probability that the generated data G (z) is real. E_x and E_z denote mean log likelihood over all real data points and synthetic data points, respectively.

$$\min_G \max_D F(G,D) = E_{X\sim Pdata(X)}\left[\log\left(D(x)\right)\right] + E_{z\sim Pz(z)}[\log\left(1-\left(D\left(G\left(z\right)\right)\right)\right] \quad (1)$$

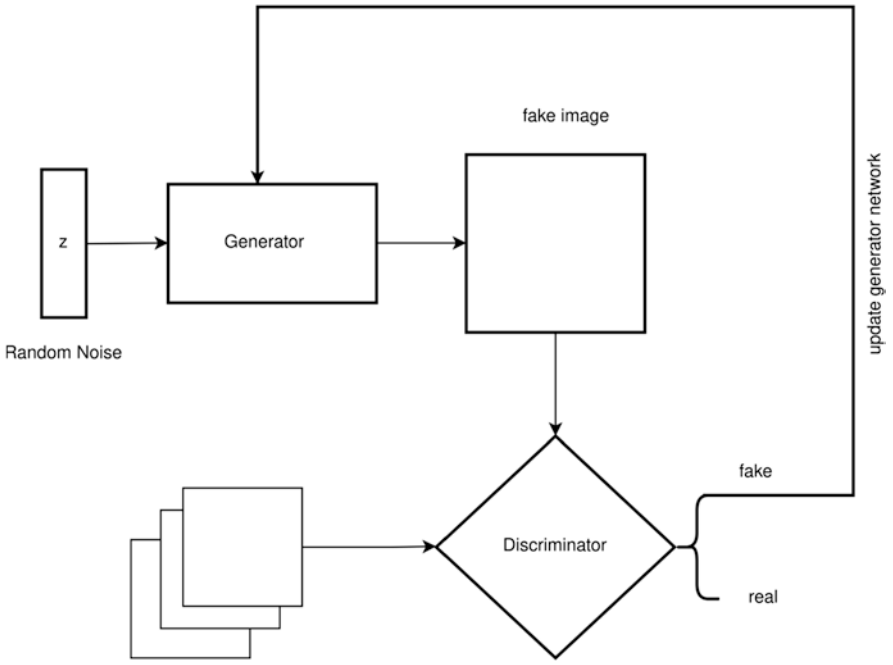


Fig. 1 Proposed Deep Convolution Generative Adversarial Network (DCGAN)

In the eq. (1) during training phase of discriminator, it tries to maximize [Log (D(x))], that is, achieving the correct labels as real or synthetic for the classification of provided data x. During training of generator, it focuses on minimizing the [Log (1-(D (G (z)))] . Here the generator cannot influence the [Log (D(x))] directly.

3.1 Generator Architecture

The architecture of the employed generator is depicted in Fig. 2. This work implemented the generator model based on transposed convolution layers. Transposed convolution layers help in reversing the down-sampling of convolution layers. This module consists of random noise z, 1 fully connected layer, and 5 transposed convolution layers. In the layer between convolutions, batch normalization and leaky ReLU activation function are utilized. The down-sampling is not recommended at generator. Transposed convolution layers have been used for up-sampling. At each layer the image size is doubled and filter size is halved. The output produced by generator module has dimensions as $256 \times 256 \times 3$.

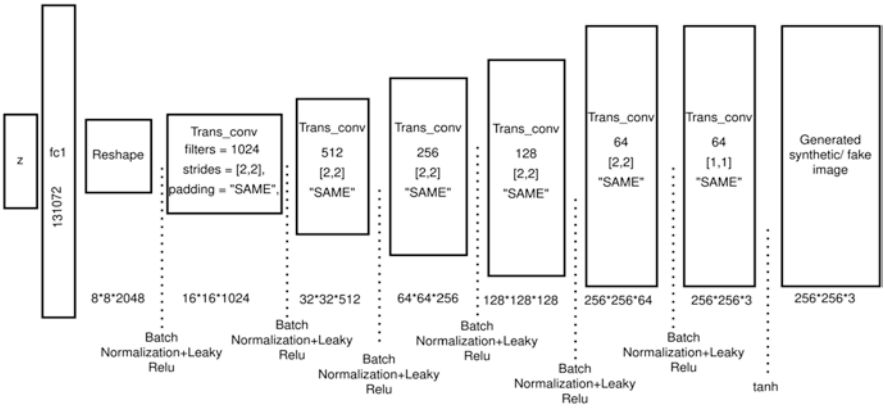


Fig. 2 Proposed Generator Architecture

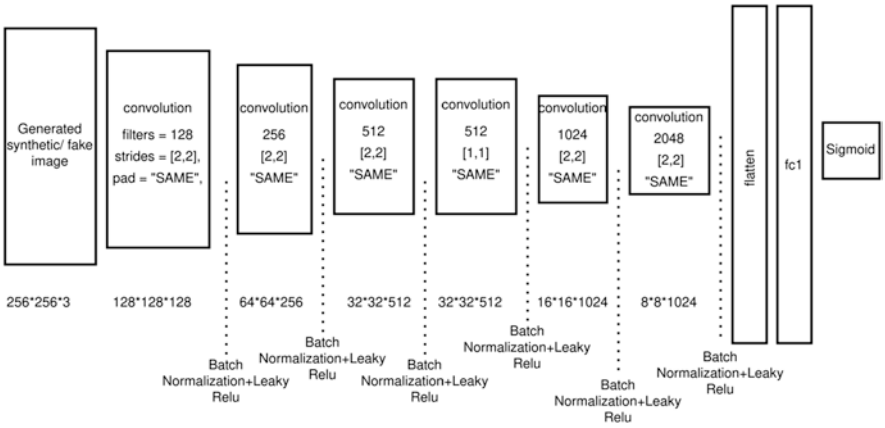


Fig. 3 Discriminator Architecture

3.2 Discriminator Architecture

The architecture of the employed discriminator is depicted in Fig. 3. The input to the discriminator is 256 × 256 × 3 either real or synthesized images. In the proposed architecture at each layer the filter size is doubled. Only strided convolution layers have been used, as down-sampling is not recommended. It uses 5 convolution layers each using leaky ReLU activation function and batch normalization. The output of the discriminator is a probability value between zero and one. The discriminator loss function is given below in eq. 2.

$$\mathrm{D\_Loss} = \mathrm{D\_Loss\_Real} + \mathrm{D\_Loss\_Fake} \tag{2}$$

D_Loss_Real is a loss when discriminator predicts real image as fake.
D_Loss_Fake is a loss when discriminator predicts fake image as real.

This work used label smoothing technique in order to facilitate discriminator to generalize better. For real images all labels are set to 1, by using label smoothing we slightly dropdown this value 1 to 0.9 or 0.95.

4 Results and Discussion

4.1 Experimental Setup

The proposed model has been built using python 3.9 and TensorFlow 2.x used to build the neural networks of generator and discriminator. The training is carried out on NVIDIA DGX-WORKSTATION; it consists of four NVIDIA Volta V100 Graphics cards each with 32GB of memory. The compute capability of these NVIDIA cards is 7.0.

4.2 Dataset Description

The GAN model is trained using data from the DRIVE fundus imaging dataset. There are 40 color fundus photos in it. These photos were taken in the Netherlands as part of a program to check for diabetic retinopathy. The photographs were taken with a Canon CR5 non-mydriatic 3CCD camera with an Field of View (FOV) of 45 degrees. Each picture has a 584 by 565 pixel resolution and three color channels.

4.3 Performance Metrics

Heusel et al. presented Frechet inception distance to evaluate the quality of GAN-generated pictures. The FID compares between the distribution of synthesized images and set of real images [6].

$$\mathrm{FID} = \left|\mu - \mu_w\right| + \mathrm{Tr}\left(\sum + \sum_w - 2\left(\sum * \sum_w\right)^{1/2}\right) \tag{3}$$

$N(\mu, \sum)$ is the multivariate normal distribution inferred using Inception-v3 characteristics calculated on real-world pictures in eq. (3). The multivariate normal distribution $N(\mu_w, \sum_w)$ is calculated using Inception v3-features computed on synthesized pictures.

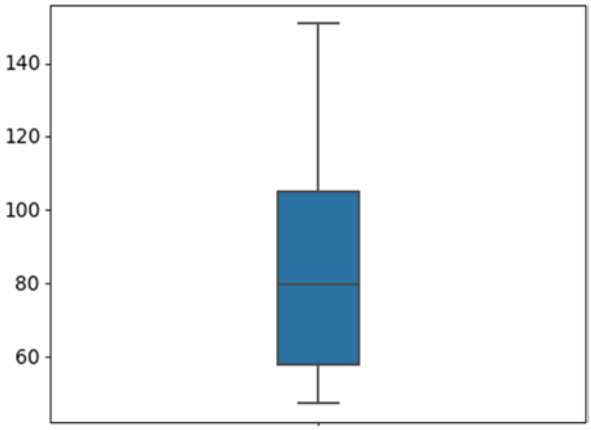


Fig. 4 Box Plot of FID Score

This work also computed seven more image similarity metrics, for the evaluation of the same the code provided by Muller et al. is used [32]. The metrics that are employed include the peak signal to noise ratio (PSNR), the structural similarity index (SSIM), the feature-based similarity index (FSIM), the signal-to-reconstruction error ratio (SRE), the spectral angle mapper (SAM), and the universal image quality index (UIQ) (Fig. 4).

4.4 Results

In this work the proposed model is trained for fundus image synthesis. The model is trained for 6000 epochs with hyper-parameter values as input image size 256 × 256 × 3, noise vector value 150, batch size of 16, discriminator learning rate as $13 \times 10^{(-6)}$, generator learning rate as $13 \times 10^{(-5)}$, and alpha value as 0.25. Adam optimizer is used. The model took approximately 18 hours to finish training. In this work we have saved images for every 5 batches and recorded corresponding epoch number and generator and discriminator loss. It is found that our model produced sharper and clearer images of size 256 × 256 × 3 after 800 epochs. Almost all images produced after 900th epoch are visually accurate and preserved details. Figure 5 shows the generator loss and discriminator loss at each epoch.

This literature makes use of FID score to perform a qualitative measure of GAN-synthesized images. To check the visual acuity of synthesized images, the FID score is computed for each batch of 20 images with the actual training dataset. Images synthesized after 900 epochs are used to measure FID score. A total of 70 image batches of 20 image each are formed and FID computations have been performed. Table 1 displays the FID score of each batch that was measured; the best FID for the batch is 48.256. The FID score pattern among GAN-generated picture batches included for evaluation. Figure 4 gives box plot distribution of FID score

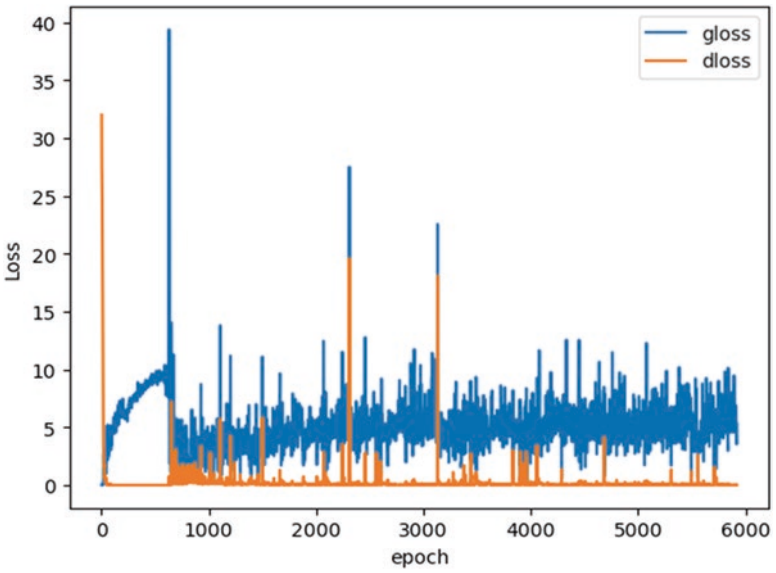


Fig. 5 The line plot of generator and discriminator loss recorded every third epoch

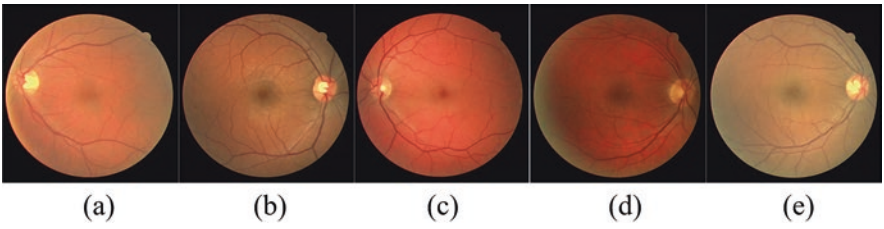


Fig. 6 Input images

along these batches. It is observed that as training proceeds, FID score tends to shift toward zero.

The generator and discriminator loss is computed at every third epoch, Fig. 5 shows the line plot of model's losses. Figure 6 shows the sample of four input images out of 40 images from DRIVE dataset. Figure 7 shows the synthesized images. It is found that the proposed model generated visually acute images in FID standards. Also it successfully captured the distribution of train dataset and produced images with varying optic disk position, vessel color, and vein network which was not present in original data (Table 2).

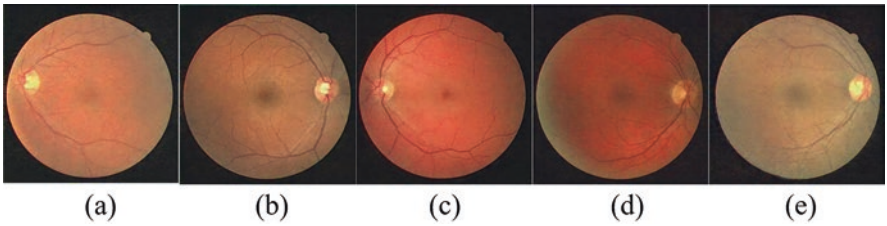


Fig. 7 GAN-generated images

Table 2 Performance metric evaluation scores for images shown in Figs. 6 against 7

Image\Metric	FID	RMSE	PSNR	SSIM	FSIM	SRE	SAM	UIQ
a	62.8273	0.002859	50.8748	0.9701	0.6794	53.3361	67.2679	0.4197
b	44.5502	0.002568	51.8060	0.9699	0.7204	51.2326	66.7308	0.5159
c	77.8374	0.002265	52.8983	0.9822	0.7064	52.9385	66.2470	0.5052
d	63.8814	0.002146	53.3639	0.9795	0.7161	50.8799	65.1703	0.4992
e	78.9070	0.003600	48.8721	0.9670	0.5832	51.4702	66.4311	0.3819

5 Conclusions

Retinal image analysis plays an important role in ophthalmology to diagnose diseases like RVO and RAO. These types of diagnoses make use of supervised algorithms hence require huge annotated datasets. Due to various factors the huge clinical imaging data is not available easily. This work proposes the GAN-based image synthesis method to generate infinite synthetic retinal images. This work avoided usage of redundant fully connected layers hence reduced the number of trainable parameters. The task of up-sampling is carried out by the usage of transposed convolution layers; it is found out that usage of transposed convolution layers gave better results. Further this approach can be extended to other imaging domains to generate synthetic images feasibly. The proposed model requires few training samples, hence can be used to generate images of specific rare diseases.

Acknowledgments We are grateful to the Computer Engineering and Information Technology department, COEP Technological University (COEP Tech.) for the provision of GPU computing facility for the computation of this work. The GPU server facility was established under the TEQIP-III (A World Bank Project).

References

1. Nakata, N. (2019). Recent technical development of artificial intelligence for diagnostic medical imaging. *Japanese Journal of Radiology, 37*, 103–108.
2. Varoquaux, G., & Cheplygina, V. (2022). Machine learning for medical imaging: Methodological failures and recommendations for the future. *npj Digital Medicine, 5*(1), 48.

3. Goodfellow, J., Pouget-Abadie, M., Mirza, B., Xu, D., Warde-Farley, S., Ozair, A. C., & Bengio, Y. (2014). Generative adversarial Neworks. *NIPS, 2014*.
4. Wang, Z., She, Q., & Ward, T. E. (2021). Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR), 54*(2), 1–38.
5. Lan, L., et al. (2020). Generative adversarial networks and its applications in biomedical informatics. *Frontiers in Public Health, 8*, 164.
6. Heusel, M., Ramsauer, H., Unterthiner, T., & Nessler, B. (2017). Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv*, 1706.08500.
7. Kim, M., et al. (2019). Deep learning in medical imaging. *Neurospine, 16*(4), 657.
8. Wang, T., et al. (2021). A review on medical imaging synthesis using deep learning and its clinical applications. *Journal of Applied Clinical Medical Physics, 22*(1), 11–36.
9. Suganthi, K. (2021). Review of medical image synthesis using GAN techniques. *ITM Web of Conferences, 37*.
10. Koshino, K., et al. (2021). Narrative review of generative adversarial networks in medical and molecular imaging. *Annals of Translational Medicine, 9*, 9.
11. Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical Image Analysis, 58*, 101552.
12. Skandarani, Y., Jodoin, P.-M., & Lalande, A. (2021). Gans for medical image synthesis: An empirical study. *arXiv preprint arXiv*, 2105.05318.
13. Bissoto, A., Valle, E., & Avila, S. (2021). Gan-based data augmentation and anonymization for skin-lesion analysis: A critical review. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
14. Andreini, P., et al. (2020). Image generation by GAN and style transfer for agar plate image segmentation. *Computer Methods and Programs in Biomedicine, 184*, 105268.
15. Chlap, P., et al. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology, 65*(5), 545–563.
16. Ghassemi, N., Shoeibi, A., & Rouhani, M. (2020). Deep neural network with generative adversarial networks pre-training for brain tumor classification based on MR images. *Biomedical Signal Processing and Control, 57*, 101678.
17. Zhu, J., Yang, G., & Lio, P. (2019). How can we make GAN perform better in single medical image super-resolution? A lesion focused multi-scale approach. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE.
18. Ma, L., et al. (2020). Combining DC-GAN with ResNet for blood cell image classification. *Medical & Biological Engineering & Computing, 58*, 1251–1264.
19. Joshi, O. S., Joshi, A. D., & Suraj, T. S. Enhancing Two Dimensional Magnetic Resonance Image Using Generative Adversarial Network. In *IEEE 9th Uttar Pradesh section international conference on electrical* (p. 2022). Electronics and Computer Engineering (UPCON).
20. Nema, S., et al. (2020). RescueNet: An unpaired GAN for brain tumor segmentation. *Biomedical Signal Processing and Control, 55*, 101641.
21. Rashid, H., Asjid Tanveer, M., & Khan, H. A. (2019). Skin lesion classification using GAN based data augmentation. In *2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE.
22. Qin, Z., et al. (2020). A GAN-based image synthesis method for skin lesion classification. *Computer Methods and Programs in Biomedicine, 195*, 105568.
23. Nie, D., et al. (2018). Medical image synthesis with deep convolutional adversarial networks. *IEEE Transactions on Biomedical Engineering, 65*(12), 2720–2730.
24. Zhou, T., et al. (2020). Hi-net: Hybrid-fusion network for multi-modal MR image synthesis. *IEEE Transactions on Medical Imaging, 39*(9), 2772–2781.
25. Sun, L., et al. (2020). An adversarial learning approach to medical image synthesis for lesion detection. *IEEE Journal of Biomedical and Health Informatics, 24*(8), 2303–2314.
26. Sravani, D. Y., & Kumar, S. P. (2022). DR-DCGAN: A deep convolutional generative adversarial network (DC-GAN) for diabetic retinopathy image synthesis. *Webology*, 19.2.

27. You, A., et al. (2022). Application of generative adversarial networks (GAN) for ophthalmology image domains: A survey. *Eye and Vision, 9*(1), 1–19.
28. Iqbal, T., & Ali, H. (2018). Generative adversarial network for medical images (MI-GAN). *Journal of Medical Systems, 42*, 1–11.
29. Shenkut, D., & Bhagavatula, V. (2022). Fundus GAN-GAN-based fundus image synthesis for training retinal image classifiers. In *2022 44th annual international conference of the IEEE engineering in Medicine & Biology Society (EMBC)*. IEEE.
30. Liang, N., et al. (2022). "end-to-end retina image synthesis based on CGAN using class feature loss and improved retinal detail loss." *IEEE. Access, 10*, 83125–83137.
31. Yu, Z., et al. (2019). Retinal image synthesis from multiple-landmarks input with generative adversarial networks. *Biomedical Engineering Online, 18*(1), 1–15.
32. Beers, A., et al. (2018). High-resolution medical image synthesis using progressively grown generative adversarial networks. *arXiv preprint arXiv*, 1805.03144.

Chest X-Ray Data Augmentation with Generative Adversarial Networks for Pneumonia and COVID-19 Diagnosis

Beena Godbin A and Graceline Jasmine S

1 Introduction

Recently, Convolutional Neural Networks (CNN) have demonstrated great outcomes on a variety of tasks when provided with appropriate amounts of training data [1–3]. Little datasets in many fields, including medical imaging, continue to be a major contributor to less CNN performance and easily prone to overfitting on training dataset. It is possible to improve the performance of CNNs by making more efficient use of the data that is already available. Several methods of augmentation, such as random rotations, flips, and the addition of a variety of noise waveforms, have been suggested [4, 5] as possible augmentation techniques. The work can be broken down into the following sections: Section 1.1 describes the works that are linked to our work. In Sect. 2, we will concentrate on the specific GAN network designs that we chose to implement. Section 3 provides a technical description of the study and the data collection used. The experiments that we carried out and the findings that we gathered are discussed in great detail in Sect. 4. In Sect. 5, we draw our conclusions, discuss potential applications for the future, and conclude our work.

1.1 Related Works

GAN model was proposed by Goodfellow [6]. Chest tomography images with high contrast and non-contrast were learned using Cycle-GAN [7], hence enabling the generation of synthetic non-contrast images of CT. This led to an improvement in

B. G. A · G. J. S (✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India
e-mail: graceline.jasmine@vit.ac.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_4

the segmentation model of various organs in computed tomography that were constructed using a U-Net based model [8]. Radford et al. [9] developed a model with deep CGAN. This was the outcome of the above mentioned. In CT scans of liver lesions and mammograms, DCGAN and CGAN (conditional-GAN) [10] significantly improved the findings. CNNs were used in order to classify the lesions employing these methodologies [11, 12]. In this research, we present a GAN model for the model of augmentation of data that is especially designed to increase the performance of another method of GAN. This model's goal is to produce better results than the original GAN model. Our model was influenced by the publication Deep Convolutional-GAN (DCGAN) [13], which came out in 2013. The use of Deep Learning models for diagnostic medical imaging has shown promising results despite the growing number of models. However, before these models can classify diseases like pneumonia and COVID-19, they need a significant amount of labeled data to learn and generalize. Several research have developed supervised methods to recognize COVID-19 markers from chest X-ray pictures using the COVID – lung xray [14] and COVIDx [15] datasets. The CNN-based COVID-NET [15] built by Wang et al. achieved an accuracy of 93.3% for multi-class classification in a test. The leftover images from each class were used to train the model. In DarkNet [16], which was created by Ozturk et al. (COVID-19 vs. No Findings), the multi-class classification (Pneumonia vs. COVID-19 vs. No Results) and binary classification. They analyzed 24 COVID-19 pictures, 100 photos of normal, and 100 images of pneumonia and reported binary classification accuracy of 98.08% and multi-class classification accuracy of 0.86%.

Karim et al. [17] proposed a deep COVID explainable model for identifying COVID from chest X-ray images. Hemdan et al. COVIDX-Net.'s [18], which is constructed of different models like VGG19, InceptionV3, and DenseNet121, was evaluated on 55 images of X-ray taken from the dataset of COVID-19 lung X-ray. The images were used to find the performance of the network. 25 COVID-19 tests came back negative, while 25 COVID-19 tests came back positive. According to what they have stated, the accuracy of each analyzed architecture ranges anywhere from 50% (InceptionV3) to 90% (VGG19 and DenseNet201). Capsule networks were utilized by Afshar et al. in order to identify COVID-19-positive instances via the COVIDx dataset. Their model was initially developed with lung X-ray photos from other datasets that were not part of COVID-19. There was an area under the Region Of Characteristics (ROC) curve (AUC) of 97%, accuracy of 96.7%, sensitivity of 91%, and specificity of 96.8%. The authors did not reveal how many photographs were taken for each category of test.

Using deep learning models, Ghoshal [19] has developed a model for detecting lung diseases more easily. In a recent paper, Afshar et al. [20] explained a capsule network based model for COVID detection. The COVID-19 chest X-ray dataset [14] and COVIDx [15] datasets were imbalanced in a latest study by DeGreve et al. [21]. The results of this study showed that these models overfit to the data and were unable to generalize to other datasets. In an effort to increase the accuracy of GAN model based networks, we investigated data augmentation approaches in light of GAN's recent success in identifying abnormalities in radiological pictures [22, 23].

On one dataset, we had normal pictures of lung X-rays and images of pneumonia, and on the other, we had normal chest X-ray images, images of COVID-19, and images of pneumonia. There are COVID-19 images in both of these datasets. ANN model for diabetes prediction was proposed by Revathi [25]. Szegedy et al. [26] developed a vision model for identifying disease. A deep learning model has been proposed by Godbin and colleagues [27, 28] for the identification of pneumonia and COVID. A trained GAN model can improve generative models' accuracy by producing X-ray pictures independent of their labels by generating new X-ray pictures. We created these new images without using the labels from the originals. We assessed the effectiveness of the GAN model based on the training of DCGAN for unknown abnormalities detection (AnoGAN) [22], and our results demonstrated enhanced classification accuracy for instances of pneumonia and COVID-19 positivity with increased ROC curve area under the receiver operating characteristic (AUC), specificity, and sensitivity. Regardless of the subject matter of the photographs that is given to analyze as input, we were able to demonstrate that our trained GAN is capable of producing new data that is unique to a certain field. Due of this, an unsupervised data augmentation was allowed to take place in the case that some of the photographs included in the dataset lacked associated labels. This was made possible as a result of the fact that the data was readily available. We demonstrated the ineffectiveness of these models in successfully augmenting data to train a generative based model when compared to our GAN for detecting pneumonia and COVID-19 images by training the same DCGAN model on the augmented data using conventional augmentation techniques and creating new data using another DCGAN. This allowed us to show that our GAN was able to successfully detect pneumonia and COVID-19 images. To achieve this, we generated additional data using a different DCGAN for the data augmentation and trained the same DCGAN model on the supplemented data. Because of this, we were able to demonstrate that our GAN is superior when it comes to recognizing COVID-19 pictures and pneumonia.

2 GAN Architecture

GANs are a particular type of framework for a generative method. A generative model is one that attempts to learn the data distribution in an implicit manner p_{data} from a dataset of sample models $x^{(1)}, \ldots, x^{(n)}$ to further create new data samples taken from the learned model distribution. We employed a technique called Deep Convolutional GAN (DCGAN), which involves using deep CNNs for the Generator (G) model and the Discriminator (D) networks. This method is made up of two neural networks, both of which undergo training at the same time. The initial network, which is referred to as the discriminator, is represented by the letter D. The discriminator is responsible for determining which samples are authentic and which ones are counterfeit. It takes in a sample of x as its input and output D(x), the likelihood of it being a genuine representation of the population.

The second network, which is referred to as the generator, is represented by the letter G. The generator creates synthetic data samples that D will, in all likelihood, regard to be genuine samples. G receives input sample models $z^{(1)}$, …, $z^{(m)}$ from a well-known simple data distribution model p_z, basically an equal kind of data distribution, and it merges G(z) to the image space of data distribution model p_g. The aim of G is to get $p_g = p_{data}$. In order to train adversarial networks, one must first optimize the value loss function of a two-player minimax game, which looks like this:

$$\min G \max D E_{x\sim pdata} \log D(x) + E_{x\sim px}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

In order to maximize D(x) for pictures with $x \sim {}_{pdata}$ and minimize D(x) for images with $x \sim {}_{pdata}$, the discriminator (D) is trained. In order to trick discriminator (D) during training such that D(G(z)) p_{data}, the generator G(z) generates pictures. The generator is therefore taught to minimize 1D(G(z)) or, alternatively, to maximize D(G(z)). Throughout the training phase, both the G (generator) and the D (discriminator) become better at their respective jobs. The generator becomes good at generating images that are more original, while the discriminator (D) becomes good at differentiating actual photos from synthetic images. Because of this, it is often referred to as "adversarial training."

2.1 Generator Architecture

As input, the generator network model receives a vector of one hundred random values selected from a normal distribution, and as output, it produces a chest X-ray picture with the dimensions 64 by 64 by 1. In order to up-sample the picture, the network design includes a dense layer that has been reconfigured into a 4 by 4 size by 512, as well as four fractionally strided convolutional layers that each have a kernel size of 4 by 4. One interpretation of a fractionally strided convolution is that it is an expansion of the pixels achieved by interpolating zeros between them. The output layer is exempt from having batch normalization done to it, but every other layer of the network does. Stabilizing the process of GAN learning and preventing the generator (G) from condensing all of the sample models into a single model may be accomplished by normalizing answers such that they have a mean of zero value and a variance value of one throughout the whole mini-batch. All of the layers make use of ReLU activation functions, with the exception of the final layer, which makes use of activation function tanh.

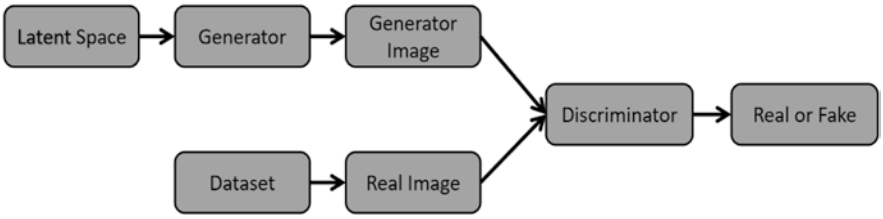


Fig. 1 Architecture of GAN

2.2 Discriminator Architecture

The discriminator (D) network receives the input picture, which is a chest X-ray measuring 64 by 64 by 1, and produces only one decision: whether or not the X-ray image in question is authentic. Figure 1 presents an illustration of the discriminator network's underlying architecture. The network has a fully connected layer in addition to its four convolution layers, each of which has a kernel size of 4 by 4. Each convolution layer receives an application of stroked convolutions in order to minimize the spatial dimensionality. Every layer of the network, with the exception of the input layer and output layers, has the batch-normalization algorithm applied to it. With the exception of the output layers, which uses the Sigmoid value function to calculate the similar probability (0–1) score of the images, leaky ReLU activation value functions are incorporated to every other layer. The activation function $f(x) = \max(0, x)$ is applicable to all layers. Moreover, dropout is applied to all of the network's layers, with the exception of the second and output levels.

2.3 Classifier

Figure.1 illustrates both the design of the classifier that we propose and the relative trainable parameter values. Because of these limited input size and datasets, CNN designs often used for medical imaging feature fewer convolutional layers than those used for other applications. Grayscale input X-rays with a fixed size of 64 by 64 and normalized within the range are sent to our classification Network (0, 1). The model is comprised of three convolutional layers, each having a kernel that is 3 by 3, batch normalization, ReLU activation function, and maximum pooling layer. The output layer consists of 2 layers that are fully connected, each of which has a dropout of 0.5 and a softmax output function applied across all 4 classes. In order to reduce overfitting of the trainset, which is especially prevalent in datasets with a small size, batch normalization and dropout have been incorporated.

2.4 Batch Size

While training a GAN, it is generally advised to avoid selecting a high batch size if at all possible. The reason for this is that when the discriminator is first being trained, it is given a huge number of instances to train on due to the bigger batch size, which might lead the generator to get overwhelmed. This can result in the network being unstable, and it is possible that the network as a whole will not converge. We opted to utilize a pretty modest batch size of 32 since, taking into consideration the size of the data we selected to train our algorithm on, it was a rather small dataset. We found that the batch size was fairly adequate for our use case, and we were able to see that the discriminator and the generator were both capable of training with stability.

2.5 Transforms of Training Samples

Before feeding the training samples to the network for the purpose of training, we might chain together a variety of different picture transformations. The following are the transformations that were used:

- *Resize*: This assists in resizing the picture that was entered to match the needed size. Because our GAN only accepts photographs, we scale the input photos to the necessary size before feeding them to the network that are 64 pixels width and 64 pixels height.
- *Random horizontal flip*: This will rotate the picture you have been provided horizontally.
- *Grayscale*: We make use of this function in order to transform the picture that was provided as input into an image with a single channel that can subsequently be sent to the network.
- *Normalize*: We use this function to normalize the picture that has been supplied to us such that both the mean and the standard deviation have unit values.

2.6 Hyperparameters

Each of the four categories of chest X-ray pictures – normal, COVID-19, bacterial pneumonia, and viral pneumonia – were each assigned to their own unique DCGANs and trained to synthesize the images of chest X-ray. This method of training for both the generator and the discriminator was performed in an iterative manner. As was noted before, we worked with small batches consisting of 32 chest X-rays each $x_1^{(1)}, \ldots, x_1^{(m)}$ for each X-ray type $l \in$ (normal, COVID-19, bacterial pneumonia, and viral pneumonia) and $n = 32$ noise data samples $z^{(1)}, \ldots, z^{(n)}$ taken from equal normal distribution between [−1–1]. The leak's incline was determined to be leak = 0.2 in

the Rectified Linear Unit (ReLU) that was being used. The weights were initially set to follow a uniform normal value with a zero-centered mean value and a 0.02 deviation. We used a technique called stochastic gradient descent (SGD) in conjunction with the adaptive moment optimizer. Adam is a gradient-aware adaptive momentum estimator that takes into consideration both the first and second gradients. These moments are controlled by the parameters 1 = 0.05 and 2 = 0.989, respectively. A learning rate of 0.0001 was maintained throughout the first stage of development of both the generator and discriminator networks.

By experimenting with several alternative combinations of the learning rates for the generator and discriminator model networks, subsequent phases assessed the convergence of the entire GAN. Finally, we decided to set the generator network's learning rate at 0.002, which was a significant increase. The section on the outcomes provides an explanation for the rationale for the same.

2.7 Evaluation Metrics

In order to do an analysis of the effectiveness of a GAN, we keep track of the training statistics listed below:

- **Loss_D**: The entire amount of the losses for all of the correct batches and all of the false batches combined, which is known as the discriminator loss (log(D(x)) + log(D(G(z)))).
- **Loss_G**: It is the total loss from the generator that is calculated here as log(D(G(z))).
- **D(x):** It is a typical depiction of the discriminator's output for the entire batch of real data. This ought to start off close to 1 and maybe converge around 0.5 as G becomes better.
- **D(G(z))**:This represents the discriminator outputs on an average basis for the fake batch as a whole. The first value, denoted by D G z1, comes from a time before D is modified, whereas the second number, denoted by D(G) z2, comes from a time after D is updated. These figures should begin close to 0, and as G becomes better, they should converge to 0.5. As a discriminator is updated, the process attempts to bring D(x) as near to 1 as possible while also bringing D(G(z)) as close as possible to 0.

On the other hand, the purpose of an update to a generator is to improve D(G(z)). This means that it attempts to deceive the discriminator into believing that the images that were produced from random noise are the real ones. To put it another way, it endeavors to deceive the discriminator into thinking that the noise-based visuals are the real ones. In a perfect world, the discriminator would not be able to tell the difference between real photographs and photoshopped ones. On the other hand, this situation is difficult to accomplish in actual life. Because there were so many cases in the initial test set (9,9,9,9), we decided to use a group of 36 photos that had an equal number of chest X-rays for each of the four classes. We chose to

take the average of each test conducted for a total of 500 iterations so that we could obtain a more accurate evaluation of the performances. A classification total accuracy metric was employed in the process of assessing the accuracy of the classification model. Additionally, for this also we computed confusion matrices, sensitivity matrices, specificity matrices, accuracy matrices, and f1 score measures for each type of lesion. The following equations offer all of these measurements for your perusal and consideration:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$F1 - \text{Score} = 2 * \frac{\text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{4}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN} + \boldsymbol{FP}} \tag{6}$$

- *Total Accuracy*: This is the total degree to which the classification corresponds to the reality of the situation.
- *Accuracy*: This is the degree to which the categorization is similar to the intended class.
- *Sensitivity*: This statistic, which is also referred to as Recall, calculates the reliability of the prediction of a class (i.e., the percentage of COVID-19 affected people who are accurately identified as having some illness) and is given by the number of samples that were rightly predicted in relation to the total amount of samples belonging to that class.
- *Specifici*ty: Or precision, this evaluates how many false positives are accurately recognized out of the total false positives (i.e., the proportion of healthy individuals who are accurately classified as not having COVID-19 and pneumonia).
- *F1Score:* It is a metric that indicates how accurate a test is. It is determined by taking into account both the sensitivity and the specificity.

Positive examples (P) are therefore instances from that category, whereas negative examples (N) are examples from the remaining three groups.

3 Materials and Methods

Python was used as the programming language to create the new model. The proposed model was accessed using three distinct scenarios: the first scenario involved testing the proposed method with four classes. In the second scenario, the suggested model was tested with three classes; in the third, the proposed model was tested with only two classes. The COVID-19 class participated in all of the test experiment settings. The validation phase and the testing phase are both components of each and every scenario. During the phase of validation, 20% of the total photos that were created will be used, but during the phase of testing, around 10% of the initial dataset will be utilized. During the entirety of the project, we made use of the open-source machine learning package called PyTorch. The Torch library served as the primary inspiration and foundation for the development of this library. It offers functions that are simple to implement and high-level interfaces, both of which are essential for projects involving machine learning and deep neural networks.

3.1 Dataset

All of our models were trained using the Kaggle COVID-19 chest X-ray dataset, which can be used at the below link on Github: https://github.com/vj2050/Transfer-Learning-COVID-19. The dataset has been thoughtfully organized and is comprised of four distinct categories: bacterial pneumonia, viral pneumonia, normal, and COVID-19. The dataset includes a total of 270 photos for training (60, 70, 70, 70), as well as 36 images for testing (9, 9, 9, 9). We followed the recommendation made by the original developers of the dataset and utilized a default train/test split. The genuine photographs from the aforementioned dataset are used as input to a Generative Adversarial Network, which then generates fake images for each of the four categories. Also, the same dataset is utilized in the training model for the classifier method.

The fundamental difference between the accuracy of the testing phase and the validation phase is that in the validation phase, the data that will be used to calculate the generalization capacity of the model or early halting will occur during the training time. This is the primary contrast between the accuracy of the testing phase and the validation phase. In contrast, the validation phase uses data to validate the accuracy of the predictions made by the model. During the testing phase, the data are utilized for a variety of objectives, including training and validating, but also for other purposes (Fig. 2).

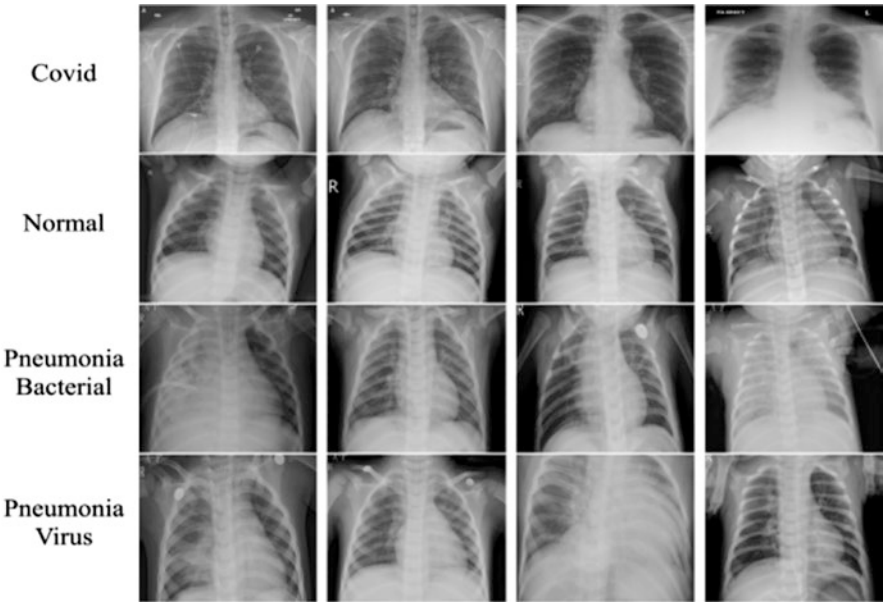


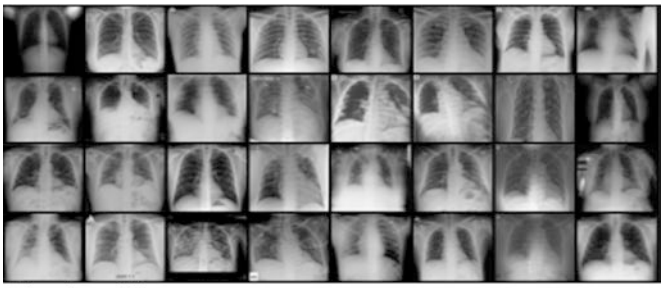
Fig. 2 Sample images

4 Experimental Details and Results

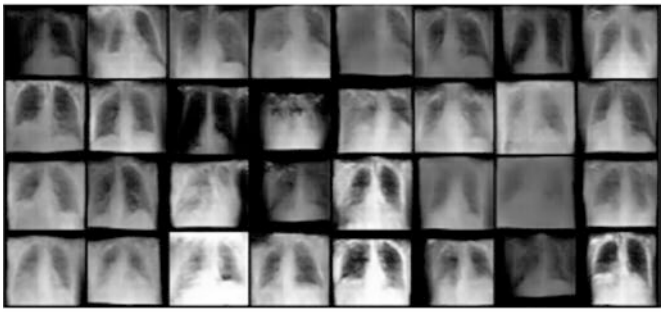
In this part of the article, we will talk about the many tests that were carried out using the various models that we trained in order to finish the mission. In this section, the results are presented for the purpose of the categorization of the chest X-ray images that are currently available. These images fall into one of four categories: COVID-19, normal, pneumonia bacteria, or pneumonia virus. We analyze the final results received while creating fake images from the GAN for the purpose of augmenting the data. We also conduct a comparison of the performance of the CNN when the traditional data augmentation approach is applied, as well as when the augmentation is carried out with the help of false pictures produced by the GAN.

4.1 Synthetic Images Generated from GAN

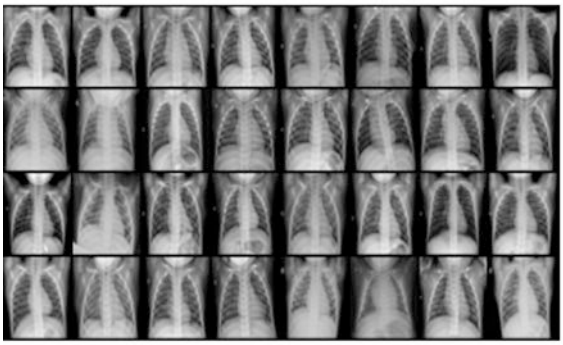
During the training process of the GAN, we utilized the COVID-19 lung X-ray dataset. For each trained network, the pertinent parameters, in particular loss D, are as follows: For the purpose of analysis, the values for Loss G, D(x), D(G(z1)), and D(G(z2) were recorded and plotted against the epoch number [24]. This was helpful in tracking the stability of the generator networks as well as the discriminator networks as the number of training epochs increased. Figure 3 is a stack of genuine

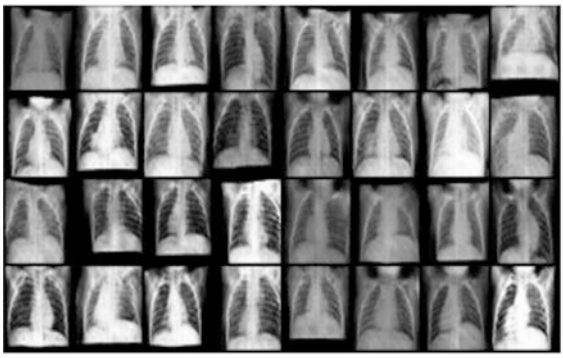


(a) real covid images



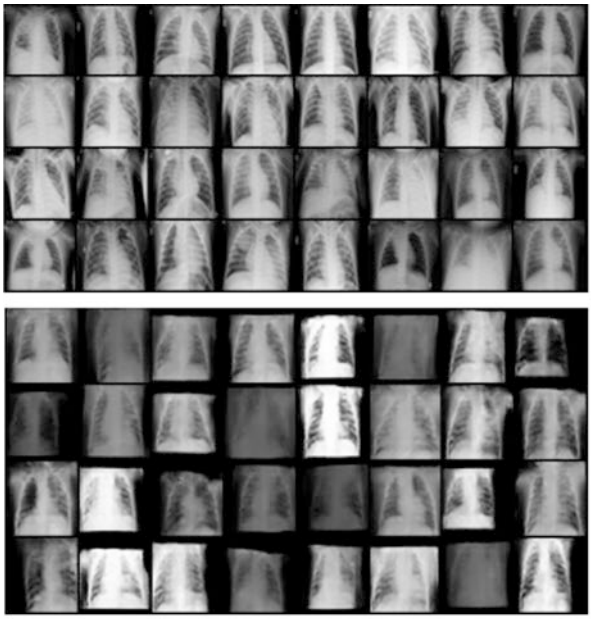
(b) fake covid images



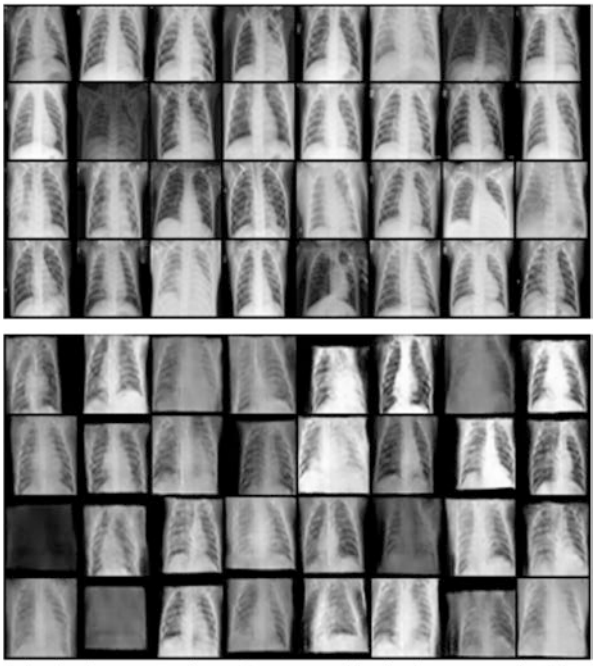


(c) real normal images (d) fake normal images

Fig. 3 Images of real (left) and fake X-ray images created by the GAN (right). (**a**, **b**) For COVID-19; (**c**, **d**) for normal; (**e**, **f**) for bacteria pneumonia; (**g**, **h**) for virus pneumonia



(e) Bacterial pneumonia real (f) Bacterial pneumonia fake



(g) Viral pneumonia real (h) viral pneumonia fake

Fig. 4.3 (continued)

X-ray pictures taken from the datasets, together with synthetic images of X-ray created by GAN model after it was trained on all of the four classes – bacterial pneumonia, viral pneumonia, normal, and COVID-19 – that are being considered. The false pictures that are displayed in Fig. 3 are one channel images that were created from a separate GAN that was trained for each of the four classes using 512 iterations. We present in the following sections a brief summary of the numerous experiments and analyses carried out when training GANs in a variety of different settings.

We trained a total of four GANs, one each for the bacterial pneumonia, viral pneumonia, COVID-19, and normal classes. Because of its slow learning rate of 0.0002, the generator network initially created synthetic images of poor quality. The discriminator was dominating the generator network, which was the primary source of the problem.

(a) Real COVID-19 images.
(b) Fake COVID-19 images.
(c) Real normal images.
(d) Fake normal images.
(e) Bacterial pneumonia real.
(f) Bacterial pneumonia fake.
(g) Viral pneumonia real.
(h) Viral pneumonia fake.

The quality of the network's output synthetic pictures was enhanced when the learning rate was adjusted to 0.002 from its previous value of 0.0001. Because of this, the generator network was able to learn in a more restrictive manner. The discriminator overtakes the generator network at lr = 0.0002 for a GAN trained on typical X-ray images, as shown in the figure. However, the generator network converges at lr = 0.002 since its loss drops down considerably more gradually over the training epochs. This is illustrated by the fact that the discriminator dominates the generator network at this value. After the discovery that a generator network that was trained with a learning rate of 0.002 performed better than other models, we carried out an investigation on the effectiveness of training GAN at various epochs. For the purpose of making a comparison, we trained a GAN using X-ray pictures of pneumonia viruses at three distinct values of epochs: 128, 256, and 512. As the number of epochs increases, it is possible to observe an improvement in the pictures' overall quality. The more epochs that pass, the more features are included in the images that are formed, which shows that the generator network's capacity to make fake images that are similar to genuine photos is also improving. Figure 4 presents a comparison of the accuracy of the traditional set of data and the synthetic dataset. The classification system performed with a sensitivity value of 82.7% and a specificity value of 90.4% using only conventional data augmentation (average). The sensitivity of the results increased to 90.6%.

Figure 5 illustrates the performance results obtained using the standard data augmentation method. Accuracy is a metric of evaluation that is utilized most frequently for classification-related activities. It is a numerical representation of the proportion of correct forecasts. We arrive at this value by determining the proportion of all of

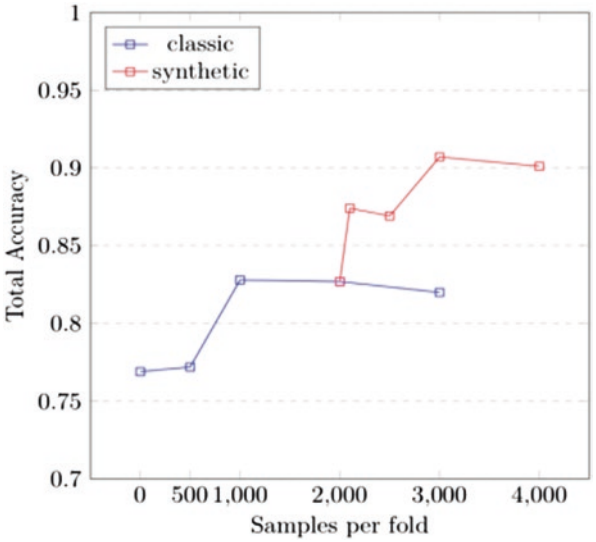


Fig. 4 Accuracy comparison between classic and synthetic dataset

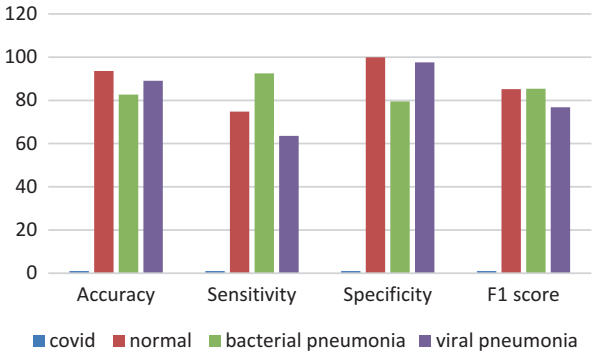


Fig. 5 Performance results on classic data augmentation

the models' predictions that were accurate to all of the other predictions. Although accuracy is the most common and well-known assessment criterion for classification, it is possible that it is not always sufficient when working with datasets taken from real life.

Criteria for categorization include the following – performance outcomes determined by traditional measures:

Precision
Recall
AUC/ROC curve

F1-score

The F-score or the F1 Score metric is used to evaluate a binary classification model, and this evaluation is based on the predictions that are supplied for the positive class. It is computed by utilizing both Precision and Recall in the process. This is a special sort of score that takes into account both your precision and your recall. The area that is beneath the ROC curve is referred known as the AUC. According to what its name suggests, AUC is used to calculate the two-dimensional area under the entire ROC curve. Along the same lines as the precision meter, the recall metric calculates the percentage of true positives that were incorrectly identified as false positives. It is possible to compute it as true positive, which refers to predictions that are actually true to the total amount of positives, regardless of whether they are accurately expected as positives or incorrectly projected as negatives (true positive and false negative).

The other significant assessment measures are depicted in Fig. 5. The accuracy of the COVID-19 class was determined to be 100%, while the f1 score for the bacterial pneumonia class was 82.7. Specificity of the viral pneumonia class was found to be 97.6 while accuracy was found to be 89.1. Figure 5 shows that the f1 score is 85.2 and that the specificity for the usual class is 99.9. The confusion matrix for the traditional data augmented dataset is displayed in Fig. 6. The prediction summary is shown as a confusion matrix. It displays the number of accurate and wrong predictions made for each class. It aids in clarifying the classes that models mistake for other classes.

Figure 7 illustrates the CNN classification performance using a dataset that has been enriched with synthetic data. It provides a comparison of the COVID-19, normal, viral, and bacterial classes based on the epochs and log-loss values. The log-loss statistic provides an indication of how well the forecast probability matches the actual or "real" value (0 or 1 if it is of binary classification). The value of the log loss

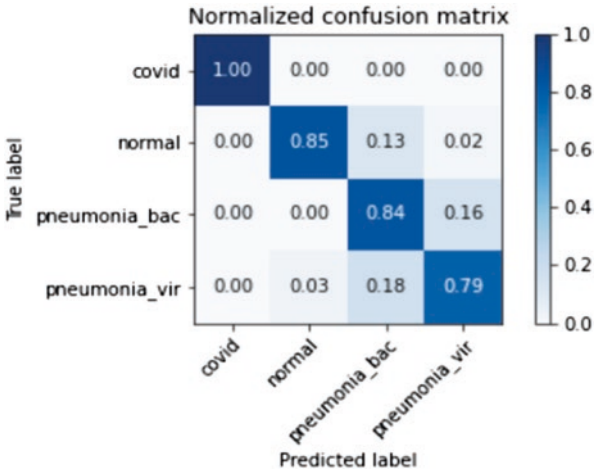


Fig. 6 Confusion matrix for classic dataset

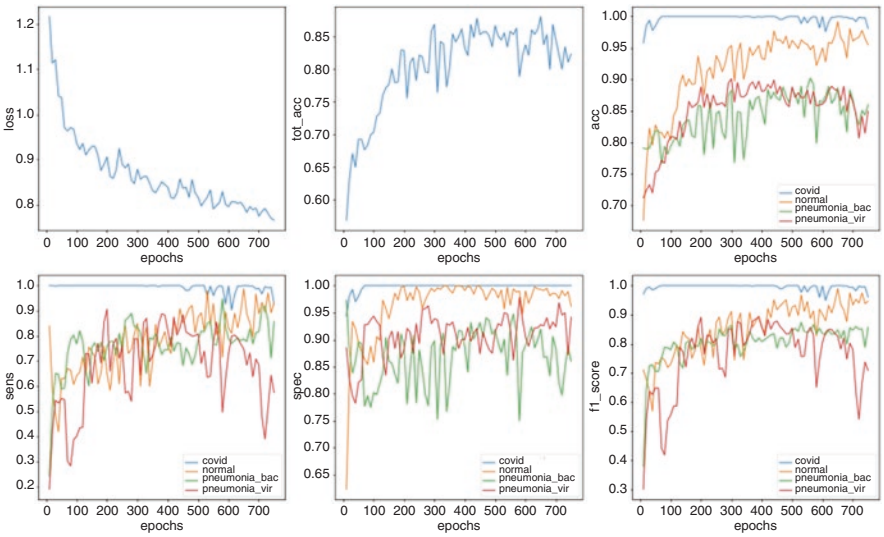


Fig. 7 CNN classification performances with classic data augmentation

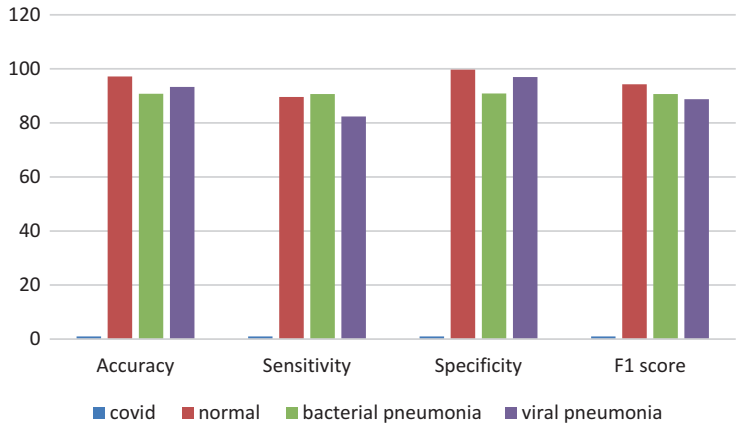


Fig. 8 Performance results on synthetic data augmentation

will go up according to the amount by which the expected probability deviates from the value that actually occurs in the experiment.

The COVID-19 class obtained a perfect score of 100%, and the bacterial pneumonia class received a score of 90.7 on the f1 scale. Specificity of 97 was reached for the viral pneumonia class, and accuracy was 93.3. As can be seen in Fig. 8, the specificity for the typical class is 99.7, and the f score is 94.3. An example of a confusion matrix can be found in Fig. 9 for the traditional data enhanced dataset. It assesses how well our classification model performs when it is asked to generate

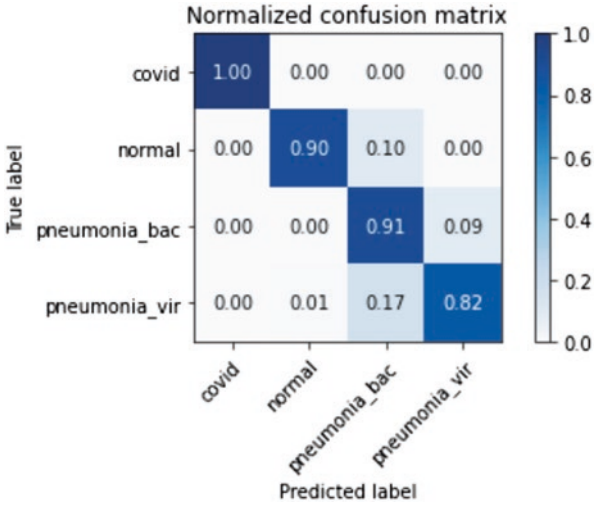


Fig. 9 Confusion matrix for synthetic augmented dataset

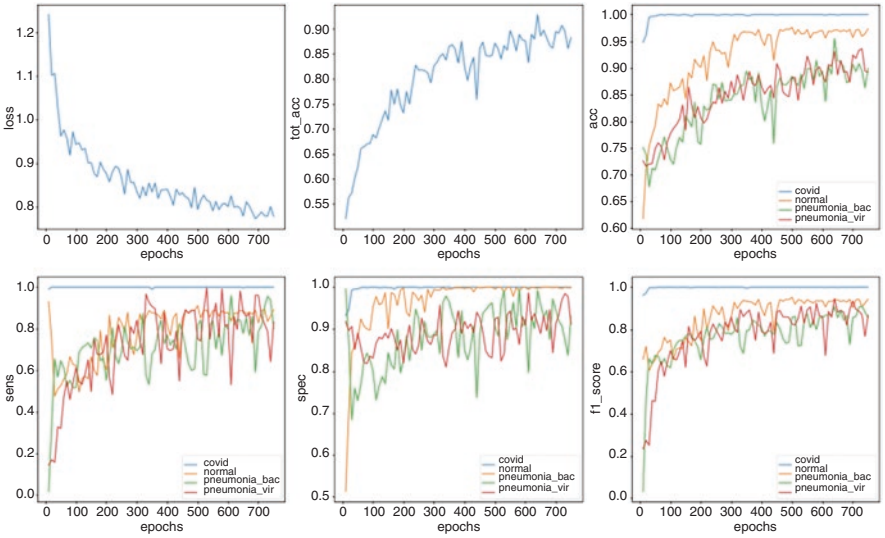


Fig. 10 CNN classification performances with synthetic (DCGAN generated) data augmentation

predictions based on test data, and it provides feedback about the overall quality of the classification model. It not only explains the mistake that the classifiers made, but it also specifies the kind of fault, such as type-I or type-II errors.

Figure 10 displays the CNN classification performance with a dataset that has been enriched with synthetic data. It presents a comparison of the classes of COVID-19, normal, viral, and bacterial organisms based on epochs.

5 Discussion and Conclusion

This study presents a GAN learning for detecting COVID-19 and pneumonia in restricted images of chest X-rays. The images were taken from a single patient. The primary impetus behind the creation of this project was the dearth of COVID-19 benchmark datasets, particularly those including chest X-ray pictures. The primary objective is to gather every conceivable image of COVID-19 and then make use of the GAN model to produce additional photos that can assist in the identification of the virus using the images of X-ray that are now accessible. A total of 270 photos from the four different classes were included in the dataset that was gathered. The categories are the contagious, normal, bacterial, and viral forms of pneumonia. The categorization and detection of the four distinct classes is a job that may be seen as being quite challenging in the absence of a large dataset. Using the developed fake images as a supplement to the original dataset would enhance classification and identification accuracy of chest X-rays. Ultimately, this would lead to a better diagnosis of lung cancer. Further research into generative adversarial architectures, which create multi-class samples simultaneously, is something that is high on our priority list. It is possible that in the near future, one of the most important things to look at will be how to use the work that was given in contexts outside of medical activities.

References

1. Krizhevsky, A., & Sutskever, I. (2012). Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–105).
2. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–34).
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–8).
4. Yu-Dong, Z., Zhengchao, D., Xianqing, C., Jia Wenjuan, D., Sidan, M. K., et al. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications, 78*(3), 3613–3632.
5. Hao, R., Namdar, K., Liu, L., Haider, M. A., & Khalvati, F. (2021). A comprehensive study of data augmentation strategies for prostate cancer detection in diffusion-weighted MRI using convolutional neural networks. *Journal of Digital Imaging, 34*, 862–876.
6. Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.
7. Veit, S., Ke, Y., Pickhardt Perry, J., & Summers, R. M. (2019). Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in CT segmentation tasks. *Scientific Reports, 9*(1), 1–9.
8. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–41). Springer.
9. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

10. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
11. Maayan, F.-A., Idit, D., Eyal, K., Michal, A., Jacob, G., & Hayit, G. (2018). Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing, 321*, 321–331.
12. Wu, E., Wu, K., David, C., & William, L. (2018). Conditional infilling GANs for data augmentation in mammogram classification. In *Image analysis for moving organ, breast, and thoracic images* (pp. 98–106). Springer.
13. Antoniou, A., Storkey, A., & Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.
14. Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., & Ghassemi, M. (2020). Covid-19 Image data collection. *arxiv:2003.11597*, https://github.com/ieee8023/covid-chestxraydataset.
15. Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports, 10*(1), 19549.
16. Tulin, O., Muhammed, T., Azra, Y. E., Baran, B. U., Yildirim Ozal, U., & Acharya, R. (2020). Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in Biology and Medicine*, 103792.
17. Karim, M., Döhmen, T., Rebholz-Schuhmann, D., Decker, S., Cochez, M., & Beyan, O. (2020). Deepcovidexplainer: Explainable covid-19 predictions based on chest x-ray images. *arXiv preprint arXiv:2004.04582*.
18. Hemdan, E. E. D., Shouman, M. A., & Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint arXiv:2003.11055*.
19. Ghoshal, B., & Tucker, A. (2020). Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. *arXiv preprint arXiv: 2003.10769*.
20. Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K. N., & Mohammadi, A. (2020). Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images. *Pattern Recognition Letters, 138*, 638–643.
21. DeGrave, A. J., Janizek, J. D., & Lee, S. I. (2020). AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence, 3*(7), 610–619.
22. Thomas, S., Philipp, S., Waldstein Sebastian, M., Ursula, S.-E., & Georg, L. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging* (pp. 146–157). Springer.
23. Saman, M., Patrik, R., & Farzad, K. (2021). Randgan: Randomized generative adversarial network for detection of covid-19 in chest x-ray. *Scientific Reports, 11*(1), 1–10.
24. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2018). Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*.
25. Revathi, M., Godbin, A. B., Bushra, S. N., & Anslam Sibi, S. (2022). Application of ANN, SVM and KNN in the prediction of diabetes Mellitus. In *2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC),* Chennai, India (pp. 179–184), https://doi.org/10.1109/ICESIC53714.2022.9783577.
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–26).
27. Godbin, A. B., & Graceline Jasmine, S. (2023). Screening of COVID-19 based on GLCM features from CT images using machine learning classifiers. *SN Computer Science, 4*(2), 1–11.
28. Godbin, A. B., & Graceline Jasmine, S. (2022). Analysis of pneumonia detection systems using deep learning-based approach. In *International conference on innovative computing, Intelligent Communication and Smart Electrical Systems (ICSES)*. IEEE.

State of the Art Framework-Based Detection of GAN-Generated Face Images

Swati Shilaskar, Shripad Bhatlawande, Siddharth Nahar, Mohammed Daanish Shaikh, Vishwesh Meher, and Rajesh Jalnekar

1 Introduction

Adobe photo editing [1] has simplified the handling of complex photos and their conversion into other high-quality images. These techniques can be used to enhance image quality [2], repair portions of images [3], or to produce complex phony images that are hard for average people to tell whether they are real or not. They can also be used to fabricate news, twist the facts, slander, and assume the identity of another person. Furthermore, such false and misrepresented material can spread swiftly via social media [4]. Various individuals can use deepfake technologies to target politicians [5], producing false information and malevolent hoaxes [6], among other things [7]. As a result, there is a growth of false pornography, hate crimes, and various forms of fraud. The victims of these crimes are suffering greatly from the harmful use of such machine learning–enabled digital technology.

Along with the aforementioned tools for altering digital photos, deep learning has made major strides in a number of other fields, such as speech recognition, image processing, and computer vision [8]. In particular, it is possible to use Generative Adversarial Networks (GANs) [9], where the discriminator and generator compete with one another, to completely generate brand-new, incredibly realistic images, movies, and voices. The main use of GANs was to produce close to real images [10] and to improve their quality [11]. Deep learning models, such as GANs, can fool users with artificially created images similar to how image editing programs can. Both humans and machine learning classifiers can be duped by a false face

S. Shilaskar · S. Bhatlawande · S. Nahar (✉) · M. D. Shaikh · V. Meher · R. Jalnekar
Electronics & Telecommunication Department, Vishwakarma Institute of Technology, Pune, Maharashtra, India
e-mail: swati.shilaskar@vit.edu; shripad.bhatlawande@vit.edu; rajesh_jalnekar@vit.edu

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_5

made by GANs [12, 13]. It can be especially problematic if they are deliberately abused for authentication of users, in addition to the generation of false information.

2 Related Work

Related work is split into three parts. The first part discusses about GANs in general, their properties, and issues. The later parts focus on GANs used for image forgery and various fake image detections techniques.

GANs [9] were first proposed in 2014 by I. Goodfellow et al. They proposed a training method to create a generative model that produces images from random noise vectors. A generative model is put up against a discriminative model that gives valuable feedback to the generator on its created images. The two models compete with each other to improve the accuracy of both resulting in a zero-sum game. GANs have been deployed in the field of [14] image processing, and image generation [15], image super-resolution, and image in painting. Modern GANs use additional methods like auto feature encoders [16] to improve the generated results and reduce distortions in the generated images. DualGAN [17] is a popular implementation of GAN where two GANs use each other's output as their input and learn to convert the information from one latent space to another latent space. They are most popularly used in image-to-image translations (converting images of one style to another). In addition to DualGAN, we have [18] StackGAN which stacks two GANs on top of each other. The first GAN outputs basic shapes and colors for the output, then the second GAN uses this image to generate details to create a more realistic image. This approach is used to solve the problem of GAN-generated images lacking details. GANs, despite their wide usage, face some challenges like [19] mode dropping (inability of generators to learn some features) and as a result they lag behind discriminators which causes the gradient of generators to vanish. In some cases, the generator tries to over-optimize the discriminator.

The performance of GANs has improved significantly from its starting years. This has led to the use of GANs in filling missing data in images. For example, the Face-Frontalization [20] framework that takes a profile photo of a person and generates a frontal image of the face by filling the missing data with generated data with the help of FI-GAN and GSP-GAN, respectively. Another implementation of GAN is found in unmasking the face [21] of people where GAN takes the image of a person with mask as input and outputs the image of that person without mask. Though the images generated by GANs are realistic they usually have some perturbation inconsistencies in the spatial distribution of RGB. In [22], a novel method for negating this limitation is proposed by balancing the distribution of R, G, and B channels in the generated images. In some cases, GANs have been wrongly used to create fake faces [23] and deep fakes. The research in [24] proposes a method for anti-face spoofing that takes a visible light image and converts it into a near-infrared image and then detects if the face is spoofed or not. Though modern techniques for

fake detection are improving, they can be fooled by simply exploiting their detection mechanism as pointed out in [25].

Fake image creation with GAN reaching realistic image production, research into fake image detection has become critical. Researchers have tried implementing simple networks from [26] LPP-SIFT for key point matching to the integration of both the host picture into non-overlapping and irregular blocks using the suggested. Much of the research points to the fact that both [27] conventional and deep learning methods can perform well in the detection of fake images but deep learning models take the lead when the image is compressed or has gone through additional processing to avoid fake image detection methods. On the other hand, Convolutional Neural Networks like [28] SFFN demonstrate the capacity to concentrate on altered facial landmarks and only require RGB images without metadata. However, [29] Dual-Order Attentive Generative Adversarial Network works with first- and second-order attenuated input fusing them with the original image data to create fused features for localizing and predicting copy-move forgery or tampering on the image. In [30], the authors exploit the most vulnerable link in GANs, then transpose convolution layer to detect inconsistencies in global information.

3 Methodology

For detection of fake images, different state-of-the-art CNN models were trained from scratch without any pretrained weights. This enabled the models to embed features specifically to detect fake images and not be influenced by weights of other prediction task, generally based on the ImageNet dataset.

3.1 Dataset

The Dataset consists of a total of 140,000 images. Fake, AI-generated images were taken from the 1 million Fake Faces Dataset developed by NVIDIA using StyleGAN [31]. These images were created in reference to Flickr Face HQ [32] real face image dataset which constitute the other half of the dataset. The images are resized to 224 × 224 and are divided into test, validation, train, and sets. A few select images from the dataset are presented in Fig. 1. Another dataset of 50,000 images was sampled from this dataset for performing five-fold cross-validation. Equal number of fake and real images were used for the same. For each fold a different set of 40,000 and 10,000 images are utilized from the same dataset and used to train and validate a new model of the same architecture. Refer to Table 1 for dataset distribution. These images span a wide variety of faces from different ethnicities, ages, and gender. This creates a dataset with less bias which helps in building a better model.

Fig. 1 Some images from the dataset. The faces in the first row are AI-generated by StyleGAN [31]. The second row consists of real images from the Flickr Face HQ dataset [32]

Table 1 Dataset distribution of fake and real images for hold-out and five-fold cross validation

Dataset for hold-out method	
Dataset class	**Number of images**
Positive images (fakes)	70,000
Negative images (real)	70,000
Train	100,000
Validation	20,000
Test	20,000
Total	140,000
Dataset for five-fold cross validation	
Positive images (fakes)	25,000
Negative images (real)	25,000
Train (per fold)	40,000
Validation (per fold)	10,000
Total (per fold)	50,000

3.2 Models Used

A total of five CNN models were used for the classification task, namely, (i) DenseNet102 [33], (ii) ResNet121 [34], (iii) MobileNetV2 [35], (iv) InceptionV3 [36], and EfficientNetB0 [37]. All of these have a unique architecture which is the basis of this comparison.

The "vanishing gradient" problem appears as CNNs get deeper, or when the number of layers in the CNN rises. DenseNets address this issue by adjusting the typical CNN architecture and streamlining the connectivity structure across layers. The name "Densely Connected Convolutional Network" comes from the fact that each layer in a DenseNet architecture is directly connected to the rest of the following layers. There are $(M\,(M+1))/2$ links for layer "M." This is visualized in Fig. 2. Each layer receives as input the feature maps from all layers that came before it. The output feature maps serve as inputs for the layers that follow. DenseNets offer a variety of benefits, these include the avoidance of issues regarding vanishing gradients, improvement in propagation of features, facilitated reuse of features, and significantly fewer parameters. A DenseNet is made up of Transition Layers and Dense Blocks, both of which contain convolutional layers. DenseNet-121 features 4 Average Pooling layers with 1 fully connected layer and 120 Convolutions. To prepare the network for the fake picture classification, a dense layer with sigmoid activation was added. The total number of trainable parameters is 6.9 million.

Deep networks suffer from a degradation issue that becomes apparent once network convergence begins. Increase in network depth causes accuracy to saturate and deteriorate substantially. To provide a solution to this problem, the deep residual learning framework ResNet was developed. ResNet uses residual mappings to fit the stacked layers. One or more layers in ResNet are bypassed using shortcut connections. Figure 3 represents the block diagram of a residual connection. The shortcut connections are used to perform identity mapping, and the outcomes are combined with those from the stacked layers. Filters of size 3×3 are used in the majority of the convolutional layers and they adhere to two straightforward rules. ResNet-101 contains 1 convolutional layer, 2 pooling layers, and 33 bottleneck blocks with 3 convolution layers each. To prepare the network for the categorization of fake images, a dense layer with sigmoid activation was added. The total number of trainable parameters is 42.5 million.

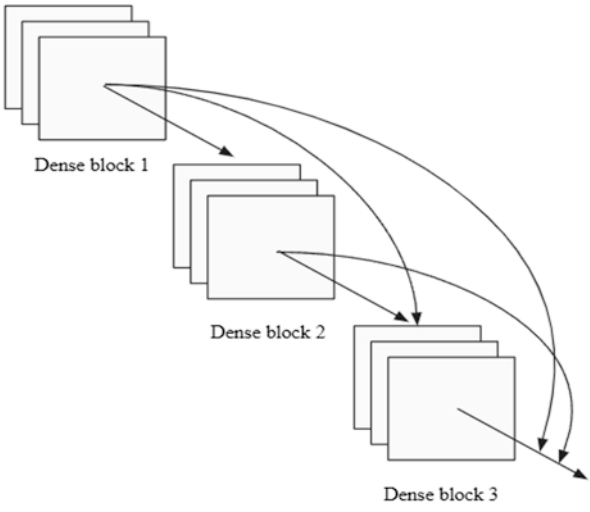


Fig. 2 The diagram displays the main part of the DenseNet architecture. Every previous layer has its output connected to the every subsequent layer [33]

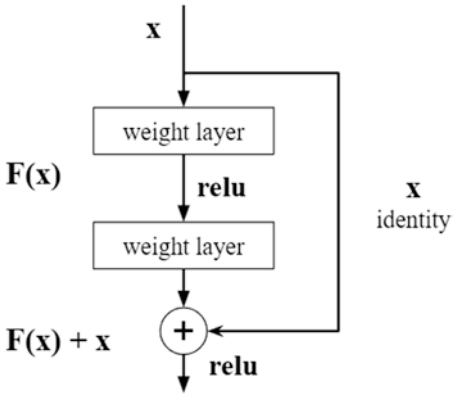


Fig. 3 The figure depicts the core component of ResNet, that is, the residual connections in ResNet for preserving the identity of the input in the case of weight degradation [34]

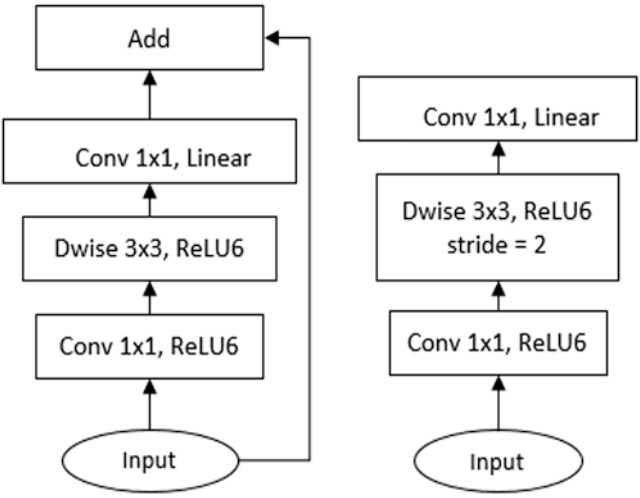


Fig. 4 Two types of bottleneck architectures of MobileNetV2, one with an inverted residual connection and stride equal to 1 and another with stride equal to 2 without the residual connection [35]

The MobilenetV2 deep learning model is an object detection and recognition model that makes use of a unique "bottleneck" architecture. This architecture employs depth wise and pointwise convolutions to maintain a comparatively low number of parameters than other state-of-the-art convolutional models. Figure 4 gives an idea of the bottleneck. The complete architecture of the model contains a normal 2-D convolution followed by 17 bottleneck layers, a 1×1 convolution, and an average pooling layer. For this particular use case, we utilized the same architecture with a change in the final layer. The number of trainable parameters was equal to 2.2 million.

InceptionV3 is another CNN network that employs pointwise convolutions. Its distinctive feature is the inception module. This module involves multiple operations on the same input and stacks the output of all these operations. This final

output serves as an input to the next layer or inception module. InceptionV3 particularly uses three types of modules or blocks, namely, module A, module B, and module C. Each of these modules has different operations but they result in a stackable final output. Module A has four parallel sequential operations that are concatenated. Three of four operations start with a pointwise operation. The third operation starts with a max pool layer, followed by a pointwise convolution. The first operation involves two 3×3 convolutions and the second parallel operation involves one 3×3 convolution. A similar presentation of modules B and C is given in Fig. 5a, b. Modules B and C use factorizing convolutions of size $1 \times n$ and $n \times 1$ instead of a whole convolution of size $n \times n$. Subsequently, this reduces the overfitting of the model while the network can go deeper to extract better features. The model in total has 42 layers, subdivided into 5 × Module A, 4 × Module B, 2 × Module C, and 2 grid reduction layers that act as regularizers. The final layer is a single-neuron dense layer with a sigmoid activation for fake image classification. The total number of trainable parameters was over 21 million.

A family of models of EfficientNets has been created by the authors to using a scalable architectural framework. All EfficientNet models are scaled from EfficientNet-B0 (see Fig. 6) using compound scaling, which includes width, depth, and resolution scaling. Any network's stem comes first, followed by architectural exploration, which is common to all eight models and the top layers. They each have seven blocks after that. The fundamental advantage of EfficientNet is the modified inverted bottleneck that is built on top of the depth wise convolution. The layer's ability to represent a solution is also simplified. As a result, the number of channels is raised to enhance overall capacity. The outcome is fewer parameters and FLOPS than other methods, but greater data flow due to the increased number of channels. GPUs are hardware accelerators built for models with high amounts of processing and where data transport is a minor component of overall performance. In the

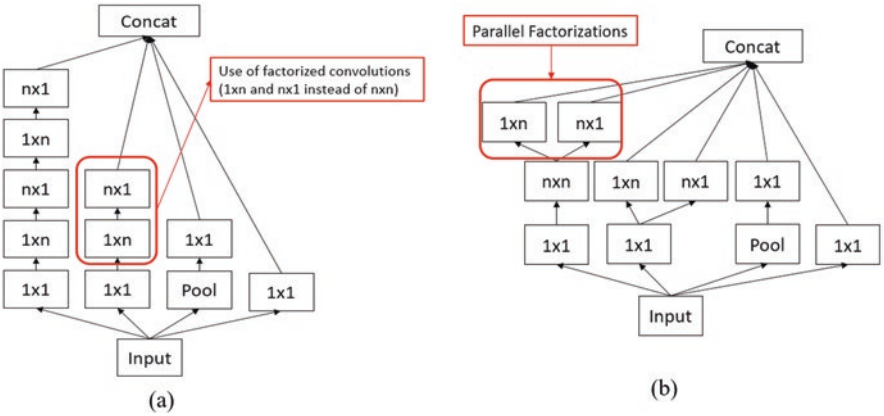


Fig. 5 (**a**) Depicts Module B of the InceptionV3 model with factorized convolutions [36] and (**b**) displays the use of parallel factorizations in Module C. These modifications serve to reduce the number of weights to be learned while preserving important information

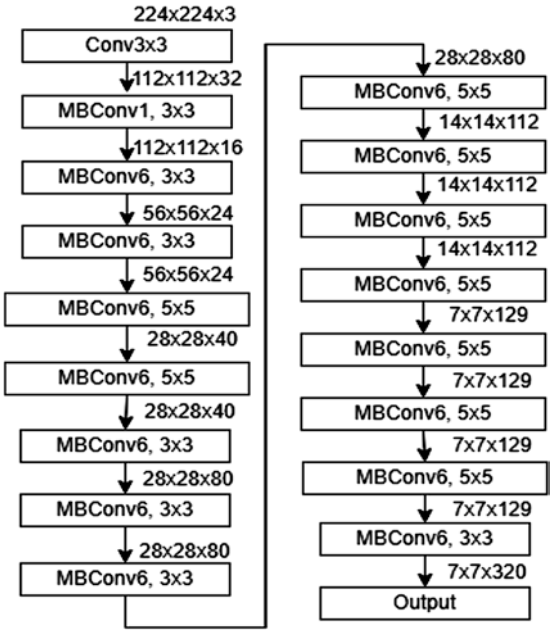


Fig. 6 This figure depicts the architecture of the EfficientNet B0 CNN model [37]

instance of EfficientNets, we have a neural network architecture that uses much less CPU while moving significantly more data than comparable networks. As a result, on hardware accelerators, EfficientNets performs badly.

3.3 Hardware and Software Setup

All models were trained on the Kaggle platform with access to a graphical processing unit (GPU). The GPU used was NVIDIA Tesla P100 with a video RAM capacity of 16 GB. The general RAM capacity of the cloud instance was 13 GB with a 2-core Intel Xeon central processing unit (CPU). All models were trained using the Tensorflow 2.6.4 framework developed by Google, with Python version 3.7.2.

3.4 Algorithms

For the hold-out method, the models were imported and were stored in a dictionary. A loop was initiated over the list and each model was trained for one iteration of the loop. Each iteration trained the specific model for 10 epochs with uninitialized weights. The trained models were then tested using a separate test dataset and thereafter the metrics were calculated. Algorithm 1 explains the process of training the

models using the hold-out method. The metrics of each model get updated to a new row with every iteration. The metrics are compared in the next section of this paper.

The five-fold cross validation method is also carried out for each of the models separately to test their effectiveness on a random sample of the whole dataset. The more the parameters the more time is required for completing the five-fold cross validation. Hence, each model was trained individually using the process explained in Algorithm 2. The scikit-learn library provides a k-fold validation function that provides indices that split the data for each fold. In this case it provides 5 set of indices. A loop is initialized and the data is split according to indices set for that particular fold and then the model is trained 5 times. This means we get 5 different models for different train and validation sets within the same dataset. The models are trained for 10 epochs. Another difference from the previous approach is that a checkpoint is set at each epoch to save the model that performs best on the validation data. It may happen that a model down performs after a certain epoch and is not able to maintain the best accuracy moving forward. Hence, this change was made.

Algorithm 1: Training and Testing the Models

Input: Model List, train data and validation data
Output: Metrics dataframe

```
//Initialize all models without pretrained weights
// Store Models in a list
Model_list = [DensNet121, Resnet102,
MobileNetV2, InceptionV3, EfficientNetB0]
for model in Model_list:
    // Initialize a sequential model
    Temp = Sequential ()
    // add model from list
    Temp.add (model)
    Temp.add (GlobalAveragePooling2D)
    //add dense layer with sigmoid activation
    Temp.add (Dense 1 with sigmoid))
    // compile the model with binary cross entropy
    // (BCE) loss and Adam Optimizer.
    Temp.compile (loss = BCE, optimizer =Adam)
    Temp.fit (train_data, validation_data, Epochs = 10)
    Y_pred = Temp.predict (test_data)
    Y_test = test_data.classes
    Report= calculate_metrics (Y_pred, Y_test)
    Dataframe = Dataframe.add (Report)
```

Algorithm 2: Five-Fold Cross Validation for One Model

Input: Dataframe 'df' containing paths to each image in the dataset.
Output: Metrics dataframe for a particular model

```
//extract indices using Kfold from sklearn
Kf = Kfold (folds=5, df)
for t_index, val_index in Kf:
    Train_data = generate data from df [t_index]
    Val_data = generate data from df [val_index]
    // Initialize the model
    Temp = Sequential (model)
    Temp.add (GlobalAveragePooling2D)
    //add dense layer with sigmoid activation
    Temp.add (Dense 1 with sigmoid))
    // compile the model with binary cross entropy
    (BCE) loss and Adam Optimizer.
    Temp.compile (loss=BCE, optimizer=Adam)
    Temp.fit (Train_data, Val_data, Epochs = 10, checkpoint = save_best_model
    Y_pred = Temp.predict (Val_data)
    Y_test = Val_data.classes
    Report= calculate_metrics (Y_pred, Y_test)
    Dataframe = Dataframe.add (Report)
```

4 Results and Discussions

Models trained using the hold-out method were tested on a separate test dataset of 20,000 images, having an equal number of real and fake images. A classification task was performed to predict if the images were real or fake. The performance metrics were derived from a confusion matrix. Accuracy along with the macro average F1-score, precision, and recall were some of the metrics utilized to evaluate the models. Macro averages were calculated by averaging the respective metrics of both classes:

$$\text{Macro}\,avg.\text{metric} = \frac{\text{Metric}_{\text{Fake}} + \text{Metric}_{\text{Real}}}{2} \tag{1}$$

where the metric can be F1-score, precision, or recall. Macro-average will be referred to as MA for simplicity (MA-F1 score, MA-Precision, and MA-Recall). Five state-of-the-art deep learning models were utilized, namely, DenseNet102, ResNet121, MobilNetV2, InceptionV3, and EfficientNetB0.

From Table 2, it is clear that InceptionV3 performed best in all metrics as compared to other models. Performance of MobilnetV2 was the lowest. It can be inferred from this that an inverted bottleneck architecture might not be suitable for capturing fine details of AI-generated facial images. The exceptional performance of InceptionV3 can be contributed to its diverse set of operations on the input image

Table 2 Comparison of detection metrics of each model for holdout method

Model	Accuracy	MA- Precision	MA- Recall	MA- F1 score
DenseNet102	81%	0.86	0.81	0.80
ResNet121	97%	0.97	0.97	0.97
MobilNetV2	61%	0.78	0.61	0.54
InceptionV3	99%	0.99	0.99	0.99
EfficientNetB0.	97%	0.97	0.97	0.97

Fig. 7 Fake images classified as fake by all the models. A probable reason for this can be the distorted backgrounds created by StyleGAN

and the concatenation of these outputs to form a new input for subsequent layers. An analysis of some images predicted by all the models was performed. The two images in Fig. 7 are fake and were classified as fake by all the models as true positives. They have visible background pattern distortions which can be easily deciphered. Figure 8a is a fake image but classified as real by four models except InceptionV3. There were particularly no background distortions in this image which may have been the cause for the misclassification. The face itself has no irregularity. This is a case of false negative. Figure 8b was a real image but classified as fake by three models except InceptionV3 and EfficientNetB0 which makes it a case of false positive. Though the image is real, the blurred hair in background seems to have caused the misclassification. Figure 8c is a real image classified as real by all models (Table 3).

Models trained using the five-fold cross validation method were trained and tested on a set of 40,000 and 10,000 images, respectively, for each fold. Five different models had five different metrics which were averaged to obtain the final result table. The metrics for a particular model in different folds of data differed by +3 or −3 than the average. The five-fold cross validation method worked well for all models, but it especially helped in improving the performance of MobileNetV2 as it converged at the seventh epoch and required no further training. A similar case was observed with DenseNet121. Further training only decreased the model's accuracy. ResNet and InceptionV3 appeared to have over fit the large training data in the hold-out method as the accuracy decreased by 5% in five-fold cross validation method.

The drawback of this research is its bias toward face images. The trained networks in this research may not work to detect other types of generated images. Hence, we call upon the creation of a diverse dataset of generated images with their

Fig. 8 Left to right, (**a**) is a case of false negative, (**b**) is a case of false positive, and (**c**) is a true negative classification

Table 3 Comparison of averaged detection metrics of each model using five-fold validation

Model	Accuracy	MA- Precision	MA- Recall	MA- F1 score
DenseNet102	93%	0.93	0.93	0.93
ResNet121	92%	0.93	0.92	0.92
MobilNetV2	84%	0.86	0.84	0.84
InceptionV3	95%	0.95	0.95	0.95
EfficientNetB0.	94%	0.94	0.94	0.94

real counterparts that can create a parallel impact, essentially like the ImageNet Dataset. A dataset of this degree can help establish the premise for new classification architectures or frameworks. This will definitely be helpful for digital forensics tasks as we move toward a future of generated data that has raised ethical issues regarding its misuse.

5 Conclusion

The spread of fake photos on social media is a rapidly growing issue. The most cutting-edge GANs are used by popular mobile and online apps to produce significant changes on human face photographs, such as gender swapping, aging, etc. Even for inexperienced users, the results are incredibly straightforward to use and extremely realistic. In this study, we evaluate how well five cutting-edge deep learning models can distinguish between legitimate and fraudulent facial photos. StyleGAN was used to generate the fake facial photos. The breakthrough StyleGAN paper makes it easier than ever to create convincing false images because it provides realistic images of the highest quality and allows for superior control and knowledge of the generated images. A dataset of 140 k images was utilized containing images generated from StyleGAN and publicly available real face images from Flickr to train the 5 models without pre-trained weights for the binary classification

task. InceptionV3 provided the best result in every metric recorded with a testing accuracy of 99%. ResNet121 and EfficientNetB0 were also found suitable for the task. MobileNetV2 was the worst performer with a 61% testing accuracy. However, this was offset by performing five-fold cross validation on the same architectures.

The good performance achieved in this study indicates that while the generated fake photos might seem realistic to the human eye, there are still enough blemishes for popular neural networks to detect these images correctly. Hence, we call upon the creation of a more diverse dataset of generated images that can mitigate the drawback of this research. This in turn will help in matters regarding misuse of generated data and how to separate the real from the fakes.

References

1. Adobe Photoshop. (2020). https://www.adobe.com/products/photoshop.html. Accessed 22 Oct 2022.
2. Jenn, M. (2022). *How to use content aware fill in Photoshop (The Easy Way)*. https://expert-photography.com/content-aware-fill-photoshop. Accessed 22 Oct 2022.
3. Ke, L., Tai, Y., & Tang, C. (2021). Occlusion-aware video object in painting. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada (pp. 14448–14458). doi: https://doi.org/10.1109/ICCV48922.2021.01420.
4. Joshua, M. (2018). *Deepfakes – Is seeing still believing?* https://expertphotography.com/content-aware-fill-photoshop. https://motherboard.vice.com/en_us/article/bjye8a/reddit-fake-porn-app-daisy-ridley. Accessed 23 Oct 2022.
5. The Guardian. (2018). https://www.bbc.com/news/av/technology-40598465. Accessed 14 Dec 2022.
6. BBC News. (2017). *Viral video deepfakes celebrities*. https://www.bbc.com/news/av/technology-50242071. Accessed 12 Oct 2022.
7. BBC News. (2017). *Fake Obama created using AI tool to make phoney speeches*. https://www.bbc.com/news/av/technology-40598465. Accessed 14 Dec 2022.
8. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).
10. Pan, Z., Weijie, Y., Yi, X., Khan, A., Yuan, F., & Zheng, Y. (2019). Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access, 7*, 36322–36333.
11. Wang, L., Chen, W., Yang, W., Bi, F., & Fei Richard, Y. (2020). A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access, 8*, 63514–63537.
12. Guarnera, L., Giudice, O., & Battiato, S. (2020). Fighting deepfake by exposing the convolutional traces on images. *IEEE Access, 8*, 165085–165098.
13. Zhang, K., Liang, Y., Zhang, J., Wang, Z., & Li, X. (2019). No one can escape: A general approach to detect tampered and generated image. *IEEE Access, 7*, 129494–129503.
14. Wang, L., Chen, W., Yang, W., Bi, F., & Yu, F. R. (2020). A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access, 8*, 63514–63537. https://doi.org/10.1109/ACCESS.2020.2982224
15. Ledig, C., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 105–114), doi: https://doi.org/10.1109/CVPR.2017.19.

16. Zheng, J., Song, W., Wu, Y., Xu, R., & Liu, F. (2019). Feature encoder guided generative adversarial network for face photo-sketch synthesis. *IEEE Access, 7*, 154971–154985. https://doi.org/10.1109/ACCESS.2019.2949070
17. Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). DualGAN: Unsupervised dual learning for image-to-image translation. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2868–2876), doi: https://doi.org/10.1109/ICCV.2017.310.
18. Zhang, H., et al. (2017). StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 5908–5916), doi: https://doi.org/10.1109/ICCV.2017.629.
19. Pavan Kumar, M. R., & Jayagopal, P. (2021). Generative adversarial networks: A survey on applications and challenges. *International Journal of Multimedia Information Retrieval, 10*, 1–24. https://doi.org/10.1007/s13735-020-00196-w
20. Rong, C., Zhang, X., & Lin, Y. (2020). Feature-improving generative adversarial network for face Frontalization. *IEEE Access, 8*, 68842–68851. https://doi.org/10.1109/ACCESS.2020.2986079
21. Luan, X., Geng, H., Liu, L., Li, W., Zhao, Y., & Ren, M. (2020). Geometry structure preserving based GAN for multi-pose face Frontalization and recognition. *IEEE Access, 8*, 104676–104687. https://doi.org/10.1109/ACCESS.2020.2996637
22. Ud Din, N., Javed, K., Bae, S., & Yi, J. (2020). A novel GAN-based network for unmasking of masked face. *IEEE Access, 8*, 44276–44287. https://doi.org/10.1109/ACCESS.2020.2977386
23. Wang, Y., Ding, X., Yang, Y., Ding, L., Ward, R., & Wang, Z. J. (2021). Perception matters: Exploring imperceptible and transferable anti-forensics for GAN-generated fake face imagery detection. *Pattern Recognition Letters, 146*, 15–22.
24. Jiang, F., Liu, P., Shao, X., et al. (2020). Face anti-spoofing with generated near-infrared images. *Multimedia Tools and Applications, 79*, 21299–21323. https://doi.org/10.1007/s11042-020-08952-0
25. Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2962–2971), doi: https://doi.org/10.1109/CVPR.2017.316.
26. Su, B., & Kaizhen, Z. (2012). Detection of copy forgery in digital images based on LPP-SIFT. In *International conference on industrial control and electronics engineering* (pp. 1773–1776), doi: https://doi.org/10.1109/ICICEE.2012.469.
27. Marra, F., Gragnaniello, D., Cozzolino, D., & Verdoliva, L. (2018). Detection of GAN-generated fake images over social networks. In *IEEE conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 384–389), doi: https://doi.org/10.1109/MIPR.2018.00084.
28. Lee, S., Tariq, S., Shin, Y., & Woo, S. S. (2021). Detecting handcrafted facial image manipulations and GAN-generated facial images using shallow-FakeFaceNet. *Applied Soft Computing, 105*, 107256.
29. Islam, A., Long, C., Basharat, A., & Hoogs, A. (2020). DOA-GAN: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *2020 IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4675–4684), doi: https://doi.org/10.1109/CVPR42600.2020.00473.
30. Mi, Z., Jiang, X., Sun, T., & Xu, K. (2020). GAN-generated image detection with self-attention mechanism against GAN generator defect. *IEEE Journal of Selected Topics in Signal Processing, 14*(5), 969–981. https://doi.org/10.1109/JSTSP.2020.2994523
31. Karras, T., Laine, S., & Aila, T. (2018). *A style-based generator architecture for generative adversarial network*s. https://doi.org/10.48550/arXiv.1812.04948
32. Flickr-Faces-HQ Dataset. (2018). https://github.com/NVlabs/ffhq-dataset, Tero Karras, *NVIDIA*. Accessed 10 Oct 2022.
33. Huang, G., Liu, Z., & Weinberger, K. Q. (2016). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).
34. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

35. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).
36. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
37. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Data Augmentation in Classifying Chest Radiograph Images (CXR) Using DCGAN-CNN

C. Rajeev and Karthika Natarajan

1 Introduction

1.1 Data Augmentation

It is a method employed in machine learning and computer vision to enlarge the size of a training dataset by implementing diverse transformations to the existing data samples. The primary purpose of data augmentation is to enhance the accuracy and generalizability of machine learning models by presenting them with a broader spectrum of variations in the input data. The structure of data augmentation is depicted in Fig. 1. Some common data augmentation techniques include:

- Flipping, rotating, and scaling images.
- Adding noise or distortions to images.
- Changing the brightness, contrast, or saturation of images.
- Randomly cropping or padding images.
- Applying geometric transformations like perspective shifts.

By applying these transformations to the training data, the machine learning model can learn to recognize and classify objects and patterns in different contexts and orientations. Data augmentation can be particularly useful in cases where the amount of available training data is limited, as it allows researchers to create more diverse and representative datasets without collecting additional samples.

C. Rajeev · K. Natarajan (✉)
School of Computer Science and Engineering, VIT AP-University, Amaravati, Andhra Pradesh, India
e-mail: rajeev.21phd7135@vitap.ac.in; Karthika.n@vitap.ac.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_6

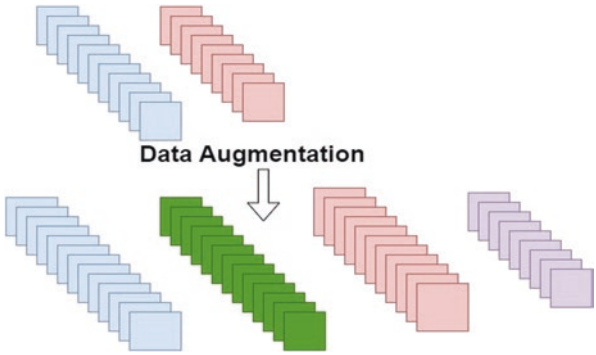


Fig. 1 Data augmentation

1.2 Augmented Versus Synthetic Data

1.2.1 Augmented Data

The term "augmented data" pertains to the generation of supplementary training data by employing different modifications or transformations to the primary data. This process involves utilizing mathematical functions or image processing techniques to the input data, creating novel versions of the data that can enhance the model's performance. The fundamental objective of data augmentation is to expand the diversity of the training set, which helps prevent overfitting and enhances the model's capacity to generalize to novel unseen data. Data augmentation can be utilized across diverse data types, encompassing but not limited to images, text, and audio.

1.2.2 Synthetic Data

Without utilizing the actual dataset, it is artificially constructed. To produce synthetic data, DNNs (Deep Neural Networks) and GANs are frequently used. The approaches for enhancement are not just for images. It may enhance text, audio, and video as well as other sorts of data. Synthetic and augmented data [1] in the context of radiological pictures are data that are not entirely produced by direct measurement from patients. With more data, machine learning models get better. Yet, there are not many open, free radiology datasets out there. Machine learning (ML) without synthetic and augmented data is difficult due to concerns about patient privacy and regulatory constraints on data use. Furthermore, certain illnesses are so uncommon that even huge datasets do not have enough examples to produce reliable machine learning algorithms. Synthetic data may be used into algorithms for artefact correction in addition to helping to create algorithms for picture identification and classification.

1.3 Why Data Augmentation Is Important Now?

There are a few of the recent developments that have made data augmentation strategies crucial.

1.3.1 Improves the Performance of ML Models

Nearly all state-of-the-art deep learning (DL) applications, such as object detection, image classification, image recognition, natural language understanding, semantic segmentation, and many others, heavily rely on data augmentation techniques [2]. The implementation of augmented data has been improving the efficiency and outcomes of deep learning models by generating novel and diverse training examples for the datasets.

1.3.2 Reduces Operation Costs Related to Data Collection

DL models may need time-consuming and expensive operations for data collecting and data labelling. By adopting data augmentation techniques to change datasets, businesses may save operating costs.

1.4 How Does Data Augmentation Work?

Data augmentation involves making a few small adjustments to the current data to create additional variants. They are accomplished by giving the dataset to transformation functions, which transform the data before creating a new dataset. Here is an example that will help you clarify the workflow of the data augmentation process which involves applying a series of transformation functions using human expertise [3]. Figure 2 depicts workflow for data augmentation. The following phases make up a typical workflow for heuristic data augmentation:

1. The data are put into the pipeline for data augmentation which contains the series of transformation operations.
2. The sequence of various augmentations that result in various iterations of each data point defines the data augmentation pipeline.

   (a) TF1 – Flipping
   (b) TF2 – Blur
   (c) TF3 – Rotation
   (d) TF4 – Skewing
   (e) TF5 – Grayscale to RGB
   (f) TFn – Brightness

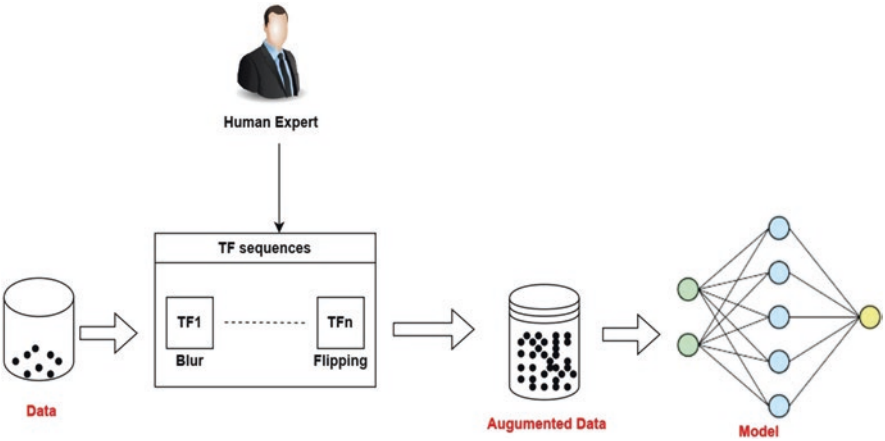


Fig. 2 Workflow of the data augmentation process

3. The image is then processed via each stage of the transformation function before being given to the data augmentation pipeline.
4. A human expert validates the boosted findings once the image has been processed.
5. The enriched findings are now ready to be used by the AI model for training after the verification procedure.

In order to work with models that will categorize photos, heuristic data augmentation is employed. In contrast to picture data, data augmentation is less common in the NLP (Natural Language Processing) discipline. The primary reason is the difficulty in automating the task of improving textual data, which stems from the intricate nature of natural language.

1.5 Advanced Techniques for Data Augmentation

To provide more varied and realistic training data, there are a number of sophisticated techniques [4] for data augmentation. Some of these methods consist of:

1. *Generative Adversarial Networks (GANs):* Synthetic data that closely mimics real-world data may be produced using GANs. A generator network is responsible for producing fresh samples, while a discriminator network determines their authenticity. Together, these two networks make up a GAN. Together, the two networks are trained, and the generator has the ability to create samples that deceive the discriminator.
2. *Autoencoders:* Neural networks with the ability to compress and decompress data are known as autoencoders. We can create new samples by randomly sampling the compressed representation and then decompressing it back to the original space after training an autoencoder on a dataset.

3. *Style Transfer:* One method for transferring one image's style to another is known as style transfer. A series of photographs can be transformed into new ones with different styles and comparable content by performing style transfer to the original images.
4. *Mixup:* A data augmentation approach called mixup involves creating new instances by linearly mixing pairs of samples from the training set. Mixup can create a new example that sits on the line linking the two original instances by merging two examples.
5. *CutMix:* With the data augmentation method known as CutMix, an image patch is randomly cropped from one image and pasted onto another image. By doing this, a new picture is produced that has the information from both the original and pasted images.
6. *Cutout*: Cutting away rectangular portions of an image at random is a data augmentation method called cutout. In addition to improving the model's capacity to generalize to new data, this drives the model to learn more robust features.

The performance and durability of machine learning models may be enhanced by using these cutting-edge strategies to provide more varied and realistic training data. They could also require more careful adjustment to prevent overfitting and be more computationally costly.

1.6 Data Augmentation in Health Care

There are several methods to use data augmentation in healthcare, including:

1. *Image Augmentation*: By using image augmentation including rotation, flipping, scaling, and cropping, medical imaging data from X-rays and MRIs can be improved [5]. This method can expand the dataset and strengthen the model's resistance to changes in the pictures.
2. *Synthetic Data Generation*: Existing datasets can be supplemented with synthetic data. Generative models, including GANs and VAEs (variational autoencoder) can be used for this, as well as simulation methods. When the current dataset is tiny or unbalanced, this strategy can be especially helpful.
3. *Text Augmentation*: Using strategies like synonym substitution, paraphrasing, and reverse translation can enhance textual data such as patient records, lab results, and clinical notes. Natural language processing models used in healthcare applications may perform better using this method.
4. *Audio Augmentation*: Time stretching, pitch shifting, and noise addition are some examples of transformations that may be used to improve audio data, such as heart and lung sounds. This method can aid in enhancing the effectiveness of illness diagnostic models.

1.7 Benefits of Data Augmentation

Some of the key benefits of data augmentation are:

1. *Improved Model Performance*: Machine learning models can perform better by being given extra data to train on thanks to data augmentation. The model may learn to spot patterns and characteristics that it might have otherwise missed by producing new data points.
2. *Better Generalization*: By integrating a wider range of diversity into the model's training, augmenting the data can enhance its capacity to generalize to novel, unseen data. This can potentially result in enhanced real-world performance.
3. *Reduced Overfitting*: When a model is unable to generalize to new data and starts identifying patterns that are specific only to the training set, it is referred to as overfitting. To mitigate this issue, the introduction of randomness and diversity through data augmentation can assist in minimizing overfitting.
4. *Reduced Data Bias*: By broadening the data's diversity, augmentation can aid in the reduction of data bias. This might be crucial in the healthcare industry because patient population bias and short datasets are common.
5. *Reduced Data Collection Costs*: It may be expensive and time-consuming to gather extensive and varied datasets. By creating new data points from current data and obviating the need for extra data gathering, data augmentation can assist in lowering these expenses.

1.8 Challenges of Data Augmentation

There are several challenges associated with this technique:

1. *Overfitting*: Data augmentation has the potential to be a potent technique in machine learning that may enhance the performance, generalization, and durability of models while lowering overfitting, bias, and data collecting costs.
2. *Computational complexity*: Applying data augmentation necessitates the use of additional computing power to create and modify fresh samples. When working with large datasets, this can result in an increase in training time and expense.
3. *Quality control*: Data augmentation may create mistakes or inconsistencies that have a detrimental impact on the model's performance. It is crucial to make sure the augmented data is of a high caliber and accurately represents the real-world data.
4. *Transformation selection*: It might be difficult to select the best data augmentation methods for a certain dataset. For particular types of data, some transformations might not be applicable or useful, while others can add unwelcome biases or distortions.
5. *Interpretability*: It may become more challenging to understand how the model behaves and makes decisions as a result of data augmentation. It might be more

difficult to understand why the model is producing particular predictions since the enhanced data might not accurately represent the distribution of the input data.

2 GANs

2.1 Introduction

GANs [6] is a type of advanced deep learning model that comprises two neural networks: the generator network and the discriminator network. The primary objective of the discriminator network is to differentiate between real and fake samples. Conversely, the generator network is designed to learn and replicate the statistical distribution of the training data. This intricate process is achieved through adversarial training, wherein both networks are simultaneously trained. Throughout the training phase, the discriminator network is exposed to authentic samples from the training data as well as counterfeit samples generated by the generator network [7]. While the generator network strives to produce samples that can trick the discriminator into classifying them as authentic, the discriminator network endeavors to correctly differentiate each sample as genuine or fake. The two networks compete in a zero-sum game, with the discriminator aiming to get better at discriminating real samples from false and the generator trying to make better fake examples. This back-and-forth between the two networks goes on until the generator generates samples that the discriminator cannot tell apart from the genuine data. The use of GANs has been extended to various tasks, such as image and video synthesis, style transfer, and data augmentation. They have become one of the most active areas of deep learning research due to their astounding performance in creating very realistic and varied samples.

2.2 Why GANs

In order to overcome the difficulty of producing realistic and varied samples of complicated data, such as images, videos, and music, Generative Adversarial Networks, or GANs, were created. Autoencoders and variational autoencoders were employed for this before to the creation of GANs, but they had limits in their capacity to produce high-quality samples with rich and diverse information [8]. Ian Goodfellow and his associates initially introduced GANs in a 2014 publication, and since then, they have grown to be one of the most well-liked and commonly applied methodologies for generative modelling. GANs are comprised of a duo of neural networks engaged in a training competition: the discriminator and the generator. The discriminator's role is to learn to differentiate between real and artificially created samples, while the generator aims to generate authentic-looking samples of the

data. Through a series of iterative training sessions, the generator progressively improves its ability to produce highly persuasive samples, fooling the discriminator. This iterative process ultimately leads to the development of top-notch generative models.

GANs have been effectively used for a variety of tasks, such as text production, music creation, style transfer, picture and video synthesis, and more. They have also been employed in fields like healthcare and finance for data augmentation and anomaly detection. In general, GANs are a significant advance in the field of deep learning and have created new opportunities for producing complicated and varied data.

2.3 Components of GANs

The primary components of GANs [9] are comprised of a generator and discriminator network.

1. *Generator*: This network creates a sample that should closely match the original data by using a random noise vector as input. The generator network, which turns random noise into an output with the same size and distribution as the real data, is often a deep neural network represented in Fig. 3.
2. *Discriminator*: This network receives a sample as input and returns a probability of whether the sample is authentic or represented in Fig. 4. A deep neural network that learns to discriminate between genuine and false data serves as the discriminator in most cases.

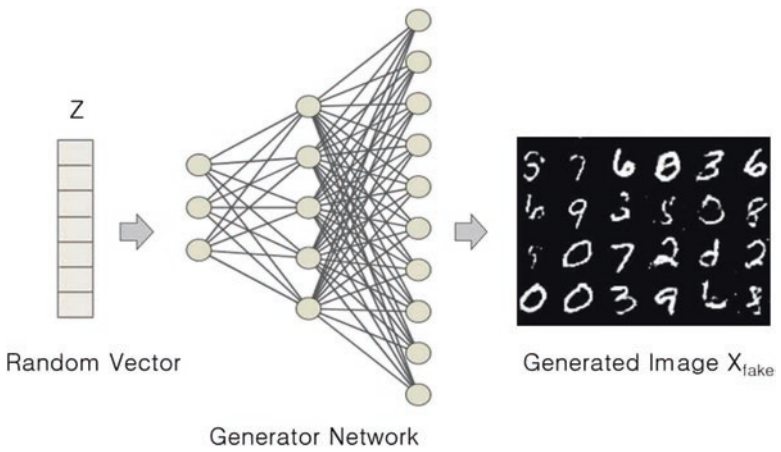


Fig. 3 Generator accepts a randomly generated vector as input and produces synthetic images of digits [10]

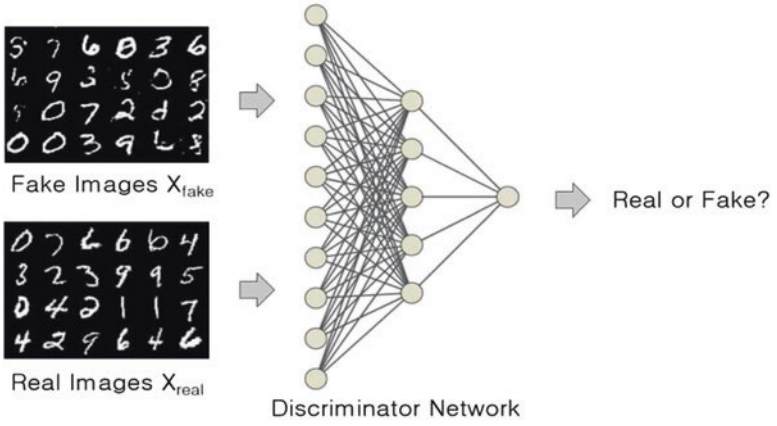


Fig. 4 Discriminator is to distinguish whether the images produced by the generator are genuine or counterfeit [10]

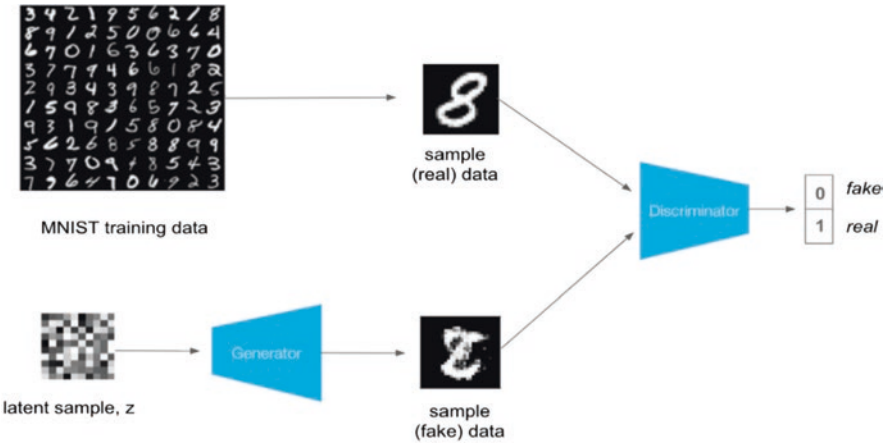


Fig. 5 Two tasks of creating synthetic data from latent space and distinguishing between genuine and fabricated data [10]

In Fig. 5, the generator network uses a latent sample as input to produce a sample that should be as similar to the real data as feasible. The generator network converts the random noise into an output whose size and distribution are identical to those of the actual data. The discriminator network receives a sample as input and returns a probability of whether the sample is authentic or not. The discriminator is trained to recognize authentic samples from false ones. The GAN architecture is given in Fig. 6.

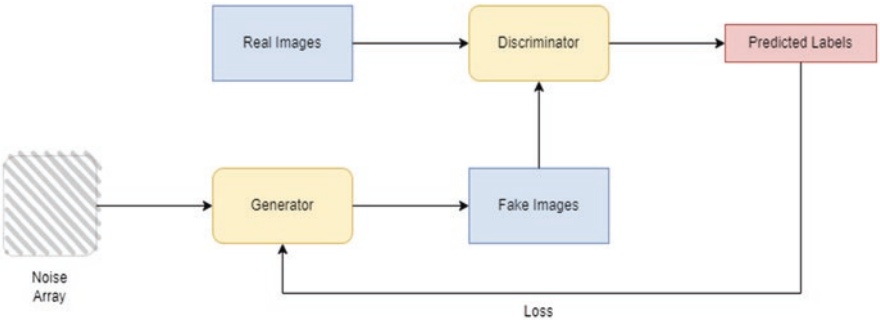


Fig. 6 GAN architecture

2.4 GAN Loss Function

Let us explore the iterative process of Generative Adversarial Networks (GANs) and how they employ the loss function [11] to minimize and maximize. To gain a deeper understanding, we will examine the following details. The generator's objective is to decrease a specific loss function, while the discriminator aims to maximize it. The loss function utilized in GANs can be represented as follows:

$$V(D,G) = E_{X\sim Pdata(x)}\left[\log D(x)\right] + E_{Z\sim PZ(Z)}\left[\log\left(1 - D\left(\left(G(z)\right)\right)\right)\right] \quad (1)$$

In Eq. (1), D(x) represents the discriminator's estimation of the probability that a real data instance x is genuine. The term Ex denotes the expected value across all real data instances. G(z) signifies the output generated by the generator when provided with noise z. On the other hand, D(G(z)) represents the discriminator's estimation of the probability that a fake instance is perceived as real. Furthermore, the term Ez indicates the expected value across all random inputs given to the generator, which can be seen as the expected value across all generated fake instances G(z).

2.5 Training and Prediction of GANs

There are six steps in the training of Generator and Discriminator and prediction of GANs. They are:

1. *Define the GAN architecture*: We define the generator and discriminator networks. The generator is commonly a neural network which generates new data by taking a random noise vector as input. On the other hand, the discriminator is a neural network that can classify real or generated data as authentic or fake.
2. *Define the loss functions*: Two distinct loss functions are defined. The generator's loss function aims to generate data that is close to the real data, while the

discriminator's loss function focuses on accurately classifying between real and generated data.
3. *Train the discriminator*: During the GAN training process, the discriminator is trained on a batch of real data as well as a batch of generated data. The discriminator is then updated to enhance its capability of differentiating between genuine and fake data.
4. *Train the generator*: To train the generator in GAN architecture, a batch of fake data is generated and passed through the discriminator. The generator is then updated to produce data that can deceive the discriminator.
5. *Repeat steps 3 and 4*: The process of training involves alternating between training the discriminator and the generator. This iteration continues until the generator generates data that is impossible to differentiate from real data.
6. *Prediction*: After completing the training process of GAN, the generator can be utilized to create new data that is analogous to the real data. The generation of new data is carried out by feeding a random noise vector to the generator, and the generator then produces new data.

2.6 Challenges Faced by GANs

GANs also face several challenges [12], including:

1. *Training instability*: GANs can exhibit an unstable nature, which may result in a phenomenon known as mode collapse. Mode collapse occurs when the generator of the GAN produces a restricted or limited range of possible outputs, failing to capture the full diversity of the desired distribution. Additionally, this can result in oscillations and vanishing gradients, making it difficult to train the model effectively.
2. *Mode collapse*: When the generator produces only a small fraction of potential outputs, it is referred to as mode collapse. This can lead to a lack of variety in the generated data. Mode collapse can occur when the discriminator becomes too dominant or the generator converges prematurely.
3. *Evaluating performance*: It can be difficult to gauge the effectiveness of GANs since conventional measures like accuracy and loss may not be appropriate. It might be challenging to judge if the created data is actually realistic or not since the generated data's quality is subjective.
4. *Overfitting*: Overfitting, when the generator becomes overly specialized in producing one sort of data and fails to provide varied data, can be a problem for GANs.
5. *Large datasets*: Working with sparse or private data can be challenging since GANs need huge datasets to train efficiently.
6. *Hyperparameter tuning*: For GANs to work well, a variety of hyperparameters must be tweaked. It might take a while and a lot of trial and error to find the ideal collection of hyperparameters.

7. *Inference speed*: When creating high-resolution photos or movies, employing GANs to generate fresh data might be time-consuming. When applying GANs in real-time applications, this can be a restriction.

It takes extensive experimentation, parameter modification, and attention to the subtleties of the GAN architecture and training procedure to overcome these obstacles. To enhance GAN performance and deal with these difficulties, researchers are always coming up with new strategies.

2.7 Different Types of GANs

There are several types of GANs, each designed for a specific task or use case. Some of the most common types of GANs are:

1. *Vanilla GAN*: A generator and a discriminator network are combined in this type of GAN, which is the earliest and most basic type. The discriminator learns to tell the difference between actual and phoney data, while the generator learns to produce realistic data.
2. *Conditional GAN*: Similar to a vanilla GAN, a conditional GAN accepts extra inputs, such as labels or characteristics, to regulate the qualities of the produced data. Common applications for this kind of GAN include style transfer and image-to-image translation.
3. *Wasserstein GAN*: The Wasserstein GAN (WGAN) is an iteration of the GAN framework that quantifies the difference between the distributions of generated and real data by utilizing the Wasserstein distance metric. WGANs can provide output of greater quality and are more stable throughout training.
4. *DCGAN*: It is a kind of GAN in which the discriminator and generator networks are both constructed using convolutional neural networks (CNNs). It has been demonstrated that DCGANs can generate high-quality pictures and are often employed for image production jobs.
5. *CycleGAN*: A CycleGAN [13] is an image-to-image translation task-specific GAN type. To make sure that the output of the generator is consistent with the input and vice versa, it employs a cycle consistency loss.
6. *Progressive GAN*: A progressive GAN is a sort of GAN that builds layers onto the generator and discriminator networks to gradually produce high-resolution pictures. Tasks requiring the production of high-resolution images frequently utilize this kind of GAN.

Numerous variants of GANs have been created, and each possesses unique advantages and limitations. Ongoing research is focused on inventing fresh and inventive GAN models to cater to diverse applications.

2.8 Steps to Implement Basic GAN

Implementing a basic Generative Adversarial Network (GAN) involves the following steps:

1. *Import the necessary libraries*: You must import the relevant deep learning libraries, such as TensorFlow or PyTorch, in order to create a GAN.
2. *Load the dataset*: The dataset from which you wish to produce synthetic data must be loaded before you can begin training a GAN. Any kind of material, including images, text, and audio, might be included here.
3. *Define the generator*: The generator network is in charge of producing synthetic data from an input of random noise. A fully connected network or a convolutional neural network are examples of deep neural network architectures that may be used to achieve it.
4. *Define the discriminator*: The discriminator network is in charge of separating authentic data from fraudulent data. An architecture for deep neural networks can also be used to implement it.
5. *Train the GAN*: To train the GAN, it is necessary to alternate between training the two networks. The generator is given random noise inputs in each cycle and utilized to create fake data. The discriminator plays a crucial role in assessing the quality of the generated data and offering feedback to the generator. This iterative process persists until the generator becomes proficient in producing high-quality fake data that can effectively deceive the discriminator.
6. *Evaluate the results*: When the GAN has been trained, you may assess the produced data's quality by contrasting it with the original data. To gauge how effectively the GAN has worked, you can use a variety of assessment criteria, including visual inspection, classification accuracy, or statistical statistics.
7. *Generate new data*: Last but not least, when the GAN has been trained, you may use it to produce fresh synthetic data by feeding it random noise inputs.

3 Augmentation of Chest Radiograph Images for Covid-19 Classification

A Deep Convolutional Generative Adversarial Network (DCGAN) is a kind of GAN in which the discriminator and generator networks are both constructed using convolutional neural networks (CNNs). DCGANs can generate high-quality pictures and are often employed for image prediction jobs.

3.1 Methodology

The approach employed involves utilizing the DCGAN-CNN technique to effectively classify CXR images into three distinct groups: normal, pneumonia, and COVID-19 [14].

3.1.1 DCGAN

DCGAN is a variant of GAN that is intended for generating images. It was first introduced in 2015 and has since become one of the most successful and commonly used architectures for generating images. DCGANs are utilized to address the issue of mode collapse, which arises when the generator becomes biased toward generating only a few outputs and fails to generate a diverse range of outputs from the dataset. For instance, consider the example of the MNIST digits dataset [15] (digits from 0 to 9). The objective is to generate all types of digits; however, sometimes the generator becomes fixated on producing only two or three digits. Consequently, the discriminator also becomes optimized to identify those specific digits only, resulting in a state known as mode collapse. Nevertheless, DCGANs can overcome this problem.

3.1.2 DCGAN Architecture

DCGAN is a neural network architecture utilized for generative modelling in several domains, such as computer vision, medical imaging, and style transfer. It is built upon the fundamental elements of GAN networks [11], including discriminators and generators. The discriminator's main purpose is to differentiate between a given sample from the synthetic or real distribution by calculating the probability value. In GAN, an optimal outcome is achieved when the probability value is near 0.5, implying that there is no differentiation between real and synthetic samples. A sample is considered genuine when its probability exceeds 0.5. By employing CNN and GANs, an unsupervised machine learning approach can proficiently model data distribution and train a generator network.

We utilized a model that employed 100×1 noise vectors denoted as "z." The network initiated from a layer of $1024 \times 4 \times 4$ and concluded with a 64×64 output layer. To enable evaluation for classification purposes, the output image underwent resizing to specific dimensions. The discriminator network analyzed actual CXR pictures and received the produced synthetic images to train the data and extract features. In the last layer, the discriminator extracted relevant characteristics and passed them to a CNN for classification. Figure 7 depicts the overall architecture of DCGAN, and the two neural networks that trained this generative model.

The network comprises a generator (G) that employs random noise Z to produce images. Gaussian noise is used as the generator's input data in GAN, representing a

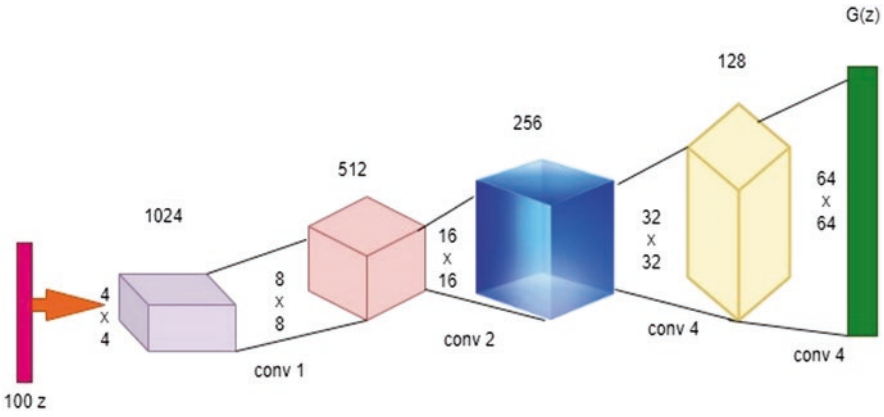


Fig. 7 General architecture of DCGAN

random point in the latent space. The discriminator (D) assesses if a picture is natural or artificial. It accepts an image x as input and outputs D(x). The outcome is based on the probability that x belongs to the real distribution. A discriminator output of 1 indicates that the picture is real, whereas a lower result indicates that it is synthetic. An upgraded DCGAN network is updated to increase GAN performance. The generator generates noisy images during the initial training stage, and after 50 epochs, the generated images resemble the original image.

3.1.3 CNN

Convolutional Neural Networks (CNNs) [16] are a specialized type of deep neural network designed specifically for tasks involving image recognition and classification. They employ a layered architecture that performs various operations, including convolutional layers, pooling layers, and fully connected layers. In the domain of CNNs, the input is typically an image or a collection of images, and the output is a prediction that identifies the objects or features present within the image. The architectural layout of a CNN is depicted in Fig. 8. The primary components of CNNs are the convolutional layers, which consist of a set of filters or kernels. These filters slide over the input image, computing a dot product between the filter and the image pixels. This process extracts different features such as edges, corners, textures, and visual patterns from the input image.

Pooling layers play a critical role in reducing the spatial dimensions of the data. They accomplish this by downsampling the output from the convolutional layers. Pooling is commonly performed using operations like maximum or average pooling, applied to a window of neighboring pixels. This downsampling operation helps decrease the size of the output volume while retaining essential features. Fully connected layers are responsible for integrating the extracted features from the convolutional and pooling layers, enabling classification or regression tasks. These layers

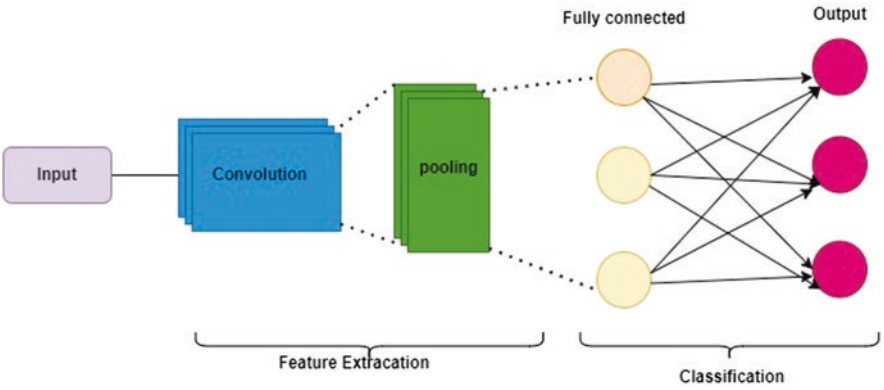


Fig. 8 CNN architecture

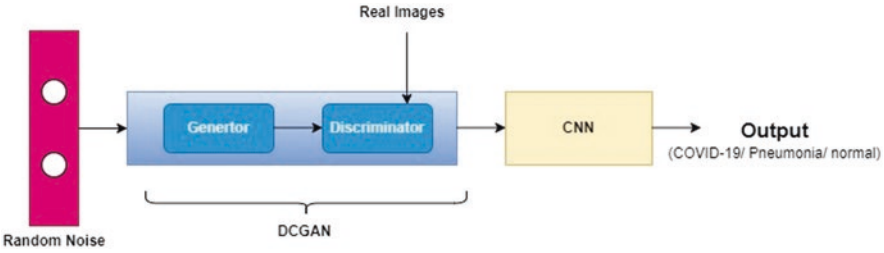


Fig. 9 DCGAN-CNN model's architecture

typically consist of multiple nodes or neurons that establish connections with every neuron in the preceding layer. This connectivity enables the network to learn intricate representations and relationships among the features, leading to accurate predictions.

3.2 DCGAN-CNN Model's Architecture

DCGAN is a multi-neural network that utilizes random noise to generate synthetic images by extracting features from input images. The network begins by extracting local features in the initial layers and subsequently utilizes these local features to extract global features. Figure 9 illustrates a block diagram representation of the DCGAN-CNN model [11].

The encoder and decoder of the DCGAN-CNN model perform downsampling and upsampling operations on the input data until reaching the bottleneck. Figure 10 provides an illustration of how the discriminator assesses the authenticity of a generated image, determining whether it is real or synthetic. The output dataset from

Fig. 10 Synthetic images generated by GAN

this evaluation is subsequently passed into the CNN classification process. On the other hand, Fig. 11. presents a flowchart representing the DCGAN-CNN model.

There are some steps to classify the COVID-19. They are as follows:

1. *Data Collection*: Collect a larger dataset of CXR images that includes: normal, pneumonia, and COVID-19 [17]. It is recommended to obtain a more comprehensive collection of images. Numerous publicly available datasets can be utilized for this purpose, including the COVIDx dataset, COVID-19 Radiography Dataset, and the Chest X-Ray14 Dataset [18].
2. *Data Pre-processing*: Pre-process the CXR dataset by resizing the images to a common size, normalizing the pixel values, and augmenting the dataset if necessary. It is also important to balance the dataset by ensuring equal representation of each class (Fig. 11).
3. *DCGAN Model*: Training a DCGAN on the CXR dataset can generate new images that resemble the original ones. These generated images can then be used to augment the original dataset and increase its diversity. It is important to use techniques like data augmentation, random rotations, and flipping to generate diverse images that capture the variations in the original dataset. Fig. 12 represents sample data augmentation of chest radiograph image [19].
4. *CNN Model*: To classify images into normal, pneumonia, and COVID-19 categories, CNN will be trained on an augmented dataset. You can use transfer learning by using a pre-trained CNN as a starting point and fine-tuning it on your augmented dataset. It is also important to use techniques like dropout and batch normalization to prevent overfitting. The final classification of the labels for normal, pneumonia, and COVID-19 is performed by the fully connected layers.

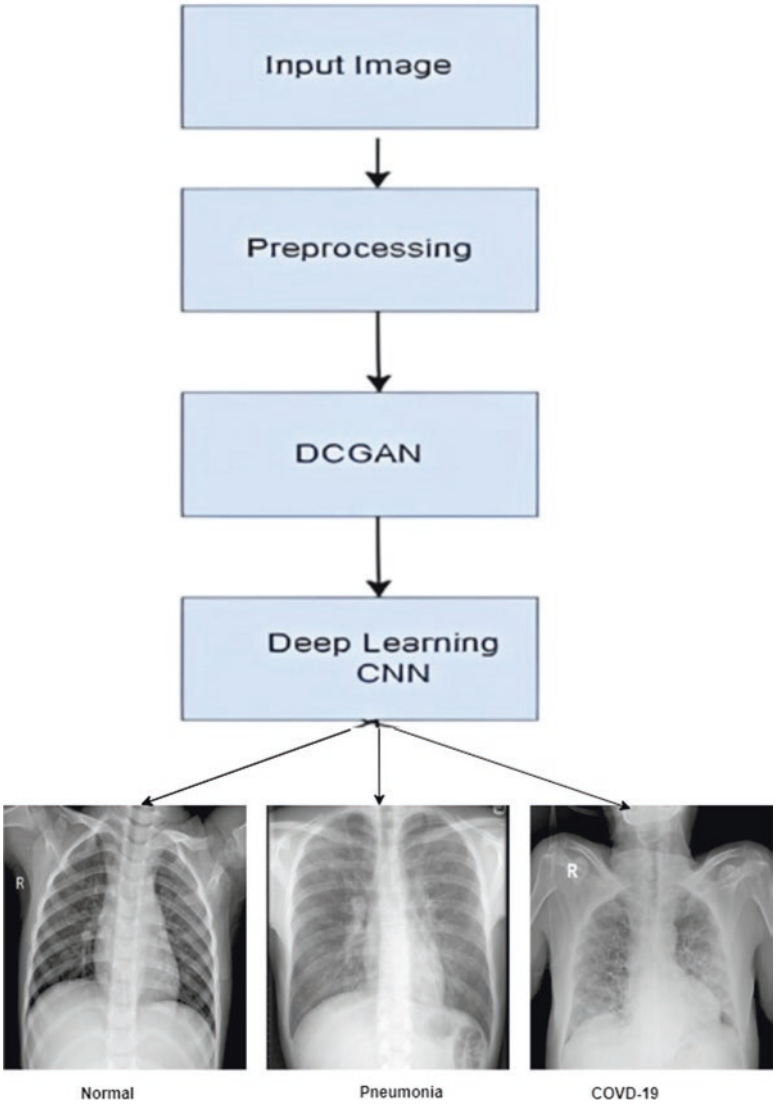


Fig. 11 Flow chart of the DCGAN-CNN model

4 Conclusion

In conclusion, the lack of diverse and sufficiently large sets of training data in medical computer vision applications remains a significant challenge, particularly due to patient privacy concerns and class imbalance. While traditional data augmentation techniques can help to increase the dataset size, they may not be sufficient to generate adequate data. However, DCGAN offers a promising solution to this problem,

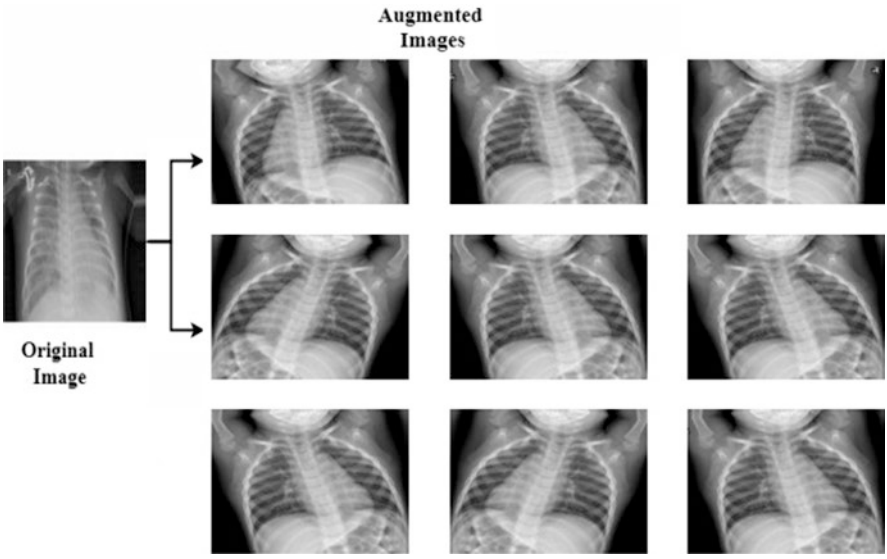


Fig. 12 Sample data augmentation of chest radiograph image

allowing the creation of clear and accurate artificial images of the under-represented class. DCGAN has great potential in the field of clinical image synthesis. The DCGAN-CNN method presents a promising solution for effective COVID-19 diagnosis. Medical image analysis researchers can leverage GAN techniques for data augmentation, thereby enhancing the performance of deep neural networks and improving the accuracy of medical diagnoses.

References

1. Sundaram, S., & Hulkund, N. (2021). GAN-based data augmentation for chest X-ray classification. In *Proceedings of KDD DSHealth. Association for computing machinery*. https://doi.org/10.1145/1122445.1122456.
2. Ciano, G., Andreini, P., Mazzierli, T., Bianchini, M., & Scarselli, F. (2021). A multi-stage GAN for multi-organ chest X-ray image generation and segmentation. *Mathematics, 9*, 2896. https://doi.org/10.3390/math9222896
3. Dilmegani, C. (2022). *A study on AI multiple*. Available on https://research.aimultiple.com/data-augmentation/. Last accessed on 26 Dec 2022.
4. Soni, P. (2022). *Analytic steps*. Available on https://www.analyticssteps.com/blogs/data-augmentation-techniques-benefits-and-applications. Last accessed on 09 Jan 2022.
5. A Complete Guide to Data Augmentation. (2022). Available on https://www.datacamp.com/tutorial/complete-guide-data-augmentation#rdl. Last accessed on Nov 2022.
6. Rashid, H., Tanveer, M. A., & Khan, H. A. (2019). Skin lesion classification using GAN based data augmentation. In *41st annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 916–919).

7. Antoniou, A., Storkey, A., & Edwards, H. (2018). *Data augmentation generative adversarial networks*. https://openreview.net/forum?id=S1AuvWRZ.
8. Agrawal, R. (2021). *An end-to-end introduction to Generative Adversarial Networks (GANs), analytics Vidya*. Available on https://www.analyticsvidhya.com/blog/2021/10/an-end-to-end-introduction-to-generative-adversarial-networksgans/. Last accessed on 20 Oct 2021.
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM, 63*(11), 139–144.
10. Aditya Sharma -Last Updated. (2021). *Introduction to Generative Adversarial Networks (GANs)*, https://learnopencv.com/introduction-to-generative-adversarial-n
11. Sharmila, V. J., & Jemi Florinabel, D. (2021). Deep learning algorithm for COVID-19 classification using chest X-ray images. *Computational and Mathematical Methods in Medicine, 2021*, 10. https://doi.org/10.1155/2021/9269173
12. Saxena, D., & Cao, J. (2021). Generative adversarial networks (GANs) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR), 54*(3), 1–42.
13. Malygina, T., Ericheva, E., & Drokin, I. (2019). Data augmentation with GAN: Improving chest X-ray pathologies prediction on class-imbalanced cases. In *International conference on analysis of images, social networks and texts*.
14. Motamed, S., Rogalla, P., & Khalvati, F. (2021). Data augmentation using generative adversarial networks (GANs) for GAN-based detection of pneumonia and COVID-19 in chest X-ray images. *Informatics in Medicine Unlocked, 27*, 100779. https://doi.org/10.1016/j.imu.2021.100779
15. Pawangfg-Last Updated. (2022). *Deep Convolutional GAN with Keras* https://www.geeksforgeeks.org/deep-convolutional-gan-with-keras/.
16. Gulakala, R., Markert, B., & Stoffel, M. (2022). Generative adversarial network based data augmentation for CNN based detection of Covid-19. *Scientific Reports, 12*, 19186. https://doi.org/10.1038/s41598-022-23692-x
17. Albahli, S. (2020). Efficient GAN-based chest radiographs (CXR) augmentation to diagnose coronavirus disease pneumonia. *International Journal of Medical Sciences, 17*(10), 1439–1448. https://doi.org/10.7150/ijms.46684
18. Ullah, Z., Usman, M., Latif, S., et al. (2023). Densely attention mechanism based network for COVID-19 detection in chest X-rays. *Scientific Reports, 13*, 261. https://doi.org/10.1038/s41598-022-27266-9
19. Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., … & Ng, A. Y. (2019). CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 590–597).

Data Augmentation Approaches Using Cycle Consistent Adversarial Networks

Agrawal Surbhi, Patil Mallanagouda, and Malini M. Patil

1 Introduction

Data is extremely important to make machine learning models to learn. If sufficient amount of data is not available then deep learning and machine learning models fail. However, in real time, most of the cases are when sufficient amount of data is not available through which we can make our models to learn. Examples of such cases are study of rare diseases, cosmic data studies where some cosmological phenomenon are rare and needs to be studied, when dataset is not balanced in terms of classes of particular type, image to image translation, etc. Other than this, there are other applications where data needs to be generated for further study like photographs of human faces, generation of new human poses, super resolution, text to image translation, video prediction, etc. Over the years some research works have been carried out in this direction. For example, Smote algorithm, oversampling, and generative models are some models which have been suggested in this direction after many researches. Out of these Smote and oversampling methods generally deal only with class imbalance in dataset. However, if it is an image dataset these methods do not work. This is where generative models take over and have been proposed.

Generative Adversarial networks is a class of unsupervised learning model where they are used to generate the fake data in order to increase the dataset. These models are famous as data augmentation tool and have been proven to successfully generate fake data where it is very difficult to identify whether it is generated data or actual data. GAN basically is a generative model that follows neural network architecture. The GANs can be understood by taking all three parts from its name:

A. Surbhi (✉) · P. Mallanagouda · M. M. Patil
RV Institute of Technology and Management, Bengaluru, Karnataka, India
e-mail: surbhiagrwal.rvitm@rvei.edu.in; mallanagoudap.rvitm@rvei.edu.in; malinimp.rvitm@rvei.edu.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_7

- *Generative* which describes the generation of data.
- *Adversarial* this provides a conducive environment for the process of training models.
- *Networks* to use neural networks to train the model.

Generative models, particularly, describe how new data generation can be done in a probabilistic manner. New data, therefore, can be generated by performing sampling from this [1]. Generative models mainly imitate the probability distribution of dataset as same as possible, then sampling is performed to generate some new distinct data points. These new data points looks like they are part of the training data. The process can be shown as in Fig. 1. In order to generate the data, training data of the entity must be available through which the model is made to learn. Generative modelling can be applied on both labeled and unlabeled data. In case of labeled data, generative models can be used to generate data belonging to each class. Whereas, for unlabeled data, this can be used to generate new data points by observing the probabilistic distribution of the existing data points. Mathematically, we can formulate generative modelling as follows:

$$\text{Generated data} = E\left[p\left(x\right)\right]\text{probabilistic estimation of observation} \tag{1}$$

$$\text{Generated data} = E[p\left(x,y\right]\text{probabilistic estimation of distribution} \tag{2}$$

Hence, GANs contain two basic and parallel trained models – one model which is used to train the model in order to generate the fake data and it is called as generator and the other to discriminate the fake data from the real data and it is called as discriminator [2].

The term adversarial indicates the competitive situation between these two discriminator and generator models. The two models try to deceive each other, that is, generator should be able to generate fake data in such a manner that it is almost impossible to distinguish between real data and fake data whereas discriminator should be able to properly determine which of the data is real and which is fake.

The term network here indicates the machine learning approach that has been used to implement the generator and the discriminator. These are mainly neural networks that can range from feed forward to convolution to very complex Unet model.

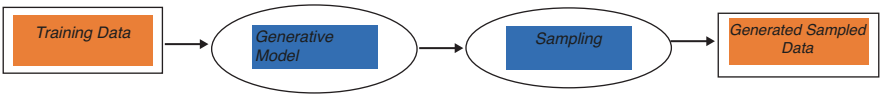


Fig. 1 Generative modeling process

1.1 GANs Application in Healthcare

GANs have their applications in various domains and different applications as given in a few example above. However, healthcare industry is the one where GANs have its multiple applications. In this chapter's subsequent discussion, the main focus will be on GAN and its subvariant cycle consistent GAN and its application mainly in healthcare industry. A small GAN operation can be discussed here. GANs till now have been successfully implemented in medical imaging where low-resolution images were a hinder for radiologists. GAN provides super-resolution to such images. For this, GANs are trained with previously collected high-resolution images. Here the images having high resolution are converted to images with low resolution and then given as input to generator. So, the generator model learns from these converted low-resolution images. The generator model then tries to enhance the resolution of these images so that the discriminator function can classify it as a real image. This continues till the loss function of the generator model with respect to discriminator model gets minimized. This is called adversarial loss.

There are challenges involved in the training of GANs. It consumes more time to train these models as they need to learn the minute characteristics or feature at a very fine level. Though training of these models is very complicated but once we achieve the accuracy of these models, they can be applied to any level. There are variety of such use cases, where some of them will be discussed in subsequent parts of the chapter, where GANs have gained popularity and can be much more helpful with its sub-variants in the health care industry.

2 Data Augmentation in Deep Learning Models

In data augmentation, already existing data is used to create a new updated dataset which is used to synthetically increase the training data. To generate this data synthetically, hence either minor adjustments are done in the dataset or with the help of deep-learning creation of new data objects can be achieved. Data augmentation is generally done for the following reasons:

- For improving the model's accuracy.
- For reducing the pre-processing in terms of labeling and cleaning the dataset.
- For increasing the training data size.
- For reducing the overfitting of a model.

Data can be augmented for above-mentioned reasons or can be synthesized. Augmentation means data is derived from the original data by applying certain transformation on them like translating, flipping, scaling, rotating, etc. This is mainly done either to increase the size of the training data or to have variety in the dataset. Whereas synthesized data is slightly different. It is artificially generated data where the original dataset may or may not be there. GANs and deep neural

networks are mainly for synthesizing the data. However, these terms are interchangeably used and techniques also overlap for generating the data in real time. In a broader sense, augmentation techniques can be applied to images, audio, text, etc. Synthetic data is mainly the oversampled augmentation which is usually added to the training dataset. Due to privacy issues however, researchers mainly are using synthesized data which as discussed is a data augmentation technique. There is, however, certain limitation of data augmentation. Few limitations are as follows:

- Performing quality assessment for enhanced dataset is difficult.
- Training of advanced techniques like GANs is challenging and their validation is still more challenging.
- It is difficult to decide on an optimal augmentation strategy.
- If any bias is present in the original data, then it will also be reflected in augmented data.

Data Augmentation Process Suppose two images of similar looking but different species are there, for example, a horse and a zebra as shown in Fig. 2. If a human being needs to identify the images they can easily tell that one is horse and other one is zebra. Humans are able to do so as they have learnt from their childhood on the basis of which feature should they say that it is a horse or it is a zebra. So, we basically identify their features and categorize them. But for a computer it is not easy to do so. If a child is once informed that an animal with black and white stripes, a short tail, etc., is a zebra, then the next time the child is able to remember that it is a zebra.

However, to make computers learn the same, many images of horses and zebras have to be given so that it can learn to distinguish between a horse and a zebra. Models like Convolution Neural Network (CNN) do not change with any kind of transformation applied on images and hence these models are very accurate in doing classifying images [3]. The concept of data augmentation is mainly based on this. A model must be invariant to any kind of transformation, that is, translation, rotation,

Fig. 2 Comparison of images

scaling, brightness, and so on. This is the key why data augmentation has success with deep learning models. The huge multiple parameters of deep learning models like CNNs enable the learning of these intricate differentiating traits through iterative "searching" through a large number of samples. As a result, the type and amount of the input dataset affect the performance of deep learning models. The main aspect to make model learn is a huge amount of data. For example, models like RESNET, BERT, and Inception- V3 need a huge amount of complex data to make the models learn properly. Unfortunately, we do not have access to a lot of data for many applications. Data Augmentation hence as discussed above gives answer for that. A general augmentation process can be shown as [4] in Fig. 3. Initially, input training data in hand is fed to the data augmentation pipeline which in turn contains a sequence of transformation steps like rotation, flipping, changing of gray scale to RGB, etc. [3]. The image is processed by each transformation with certain probability. Once processed, these images are validated by a human expert. If passed by the human expert, the training data is augmented and this augmented data is fed to the deep learning model for further processing.

An important application of data augmentation is Model Patching. Model patching is most of the times required post deployment of a machine learning model. There might be cases where an ML model may not perform well, that is, make wrong predictions after being deployed. Such situations are needed to be handled. This situation is dealt with by using model patching [5]. Practitioners have seen, for instance, that classifiers gave good accuracies for benign skin lesion images (on the ISIC skin cancer detection dataset [4]) with visible bandages than they are on benign images with no bandage. This is a subgroup problem. Theoretically, we would train a classifier to be invariant to these properties while generating predictions by automatically learning the features that distinguish the subgroups of a class. For this model patching provides an implementable solution through augmentation. For this it works in two stages:

Inter-subgroup transformation learning: Identify characteristics that set subgroups apart within a class and become familiar with the transformations that exist between them. These modifications alter the subgroup identity of an example while keeping the class designation.

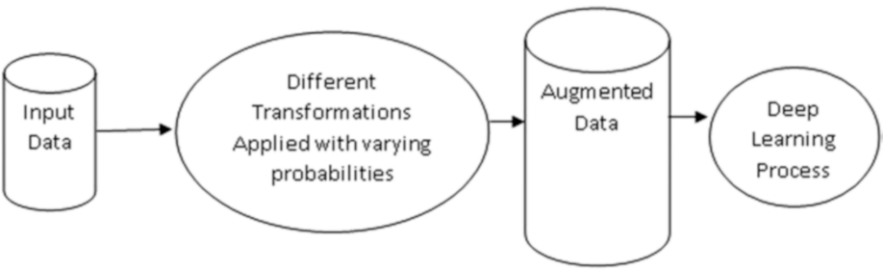


Fig. 3 Augmentation process

Training of the model for patching: Use the transformations as data augmentations which are under control to change the subgroup characteristics, allowing the classifier to be more resilient to their variance.

Modelpatching, a two-stage approach to enhancing resilience, encourages the model to be invariant to subgroup changes and concentrate on class information shared by subgroups. Model patching combined with CycleGANs have shown great performance that will be discussed in a later section in this chapter. Model patching is emerging as a field that could solve the main issue in safety-critical systems, such as healthcare (e.g., enhancing models to create artifact-free MRI scans) and autonomous vehicles (e.g., enhancing perception models that may perform poorly on erratic objects or a variety of driving conditions). As shown in Fig. 4 [4], the vanilla model which was trained on skin cancer dataset was not performing properly to differentiate images with or without bandages for malignant cancer. Selvaraju et al. [4] showed that vanilla model ambiguously relates the colored spot with benign skin lesions. However, it was correctly classified using model patching.

3 Data Augmentation in Healthcare

There are various application of data augmentation as discussed above. However, this chapter focuses on data augmentation's application in healthcare. Why this technique is required in healthcare is a big question and what it can do is the other. So, here we present some of the studies on the need of data augmentation in the healthcare domain. Data has long been a focus in the healthcare industry. In an endeavor to better the health of their patients, doctors and specialists examine photographs, patient data, and medical literature. With the introduction of electronic medical records (EMRs), networked devices, and, more recently, the Internet of Things, healthcare has become more and more digital throughout time.

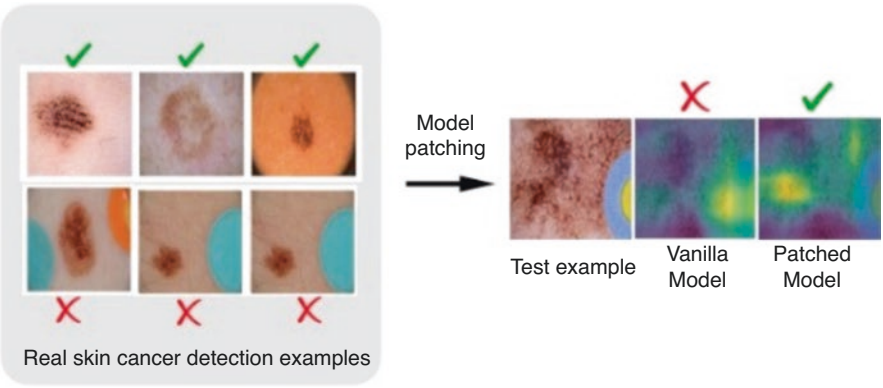


Fig. 4 Model patching of images of malignant lesion with and without colored bandage

Medical imaging datasets necessitate time-consuming and costly labeling. Before performing the analysis of data of the dataset its validation has to be done by a person having expertise in that domain. Then proper and accurate machine learning algorithms can be trained. For example, we can employ random transformation techniques like zooming, color space transformation, stretching, and cropping to enhance the model performance in the case of pneumonia classification. But one must exercise caution when using some augmentations because they may have the opposite effect. For the X-ray imaging dataset, for instance, random rotation and reflection along the x-axis are not advised. The following discussed techniques of data augmentation have been found as per [6].

3.1 Basic Augmentation Techniques

Some of the very basic augmentation techniques that have been used are as follows:

Geometric Transformations Goel et al. [5] Transformation techniques like affine such as scaling, translating, rotating, reflecting, shearing, and perspective transforms including skew are some of the more general geometric transformations which have been used so far in multiple works in the medical area to enhance the image quality.

Cropping In this technique, patches from an existing image are randomly chosen and dataset is then expanded by including these random patches once again. To address the class imbalance, mainly this technique is used. In [7], technique has been used on dental radiography. To determine which cropping approaches are best for identifying dental image objects, researchers in this study tested existing cropping methods with image object data utilizing periapical techniques on human dental pictures.

Occlusion This is a phenomenon where the visibility of an image is not proper as another image is in its path [4]. This is a critical problem in image processing which needs to be addressed. Techniques were needed which make such study easy. In the medical field, augmented techniques are used in such cases. For example, in dentistry one might be required to study a tooth which is occluded by another tooth. Augmented techniques support in such cases. Similarly, during COVID-19 it helped to study facial issues of people without removing their masks.

Some other basic augmentation techniques are noise injection, combination of images, intensity operations, etc.

3.2 Deformable Augmentation Techniques

There are cases when basic augmentation techniques do not give satisfactory result or vision of images. In those cases, some advanced augmentation techniques or image enhancers can be used. There are many such techniques like randomized displacement field, spline interpolation, and deformable image registration. In order to guarantee that the resulting augmentations are clinically plausible, the scale of distortion is often restricted by thresholds or limits as specified by the user. By using such techniques, one may also help clinical scans simulate a variety of feasible variations, such as tissue distortions or antiquity of images brought on by patient movement. In image registration process, there is an image called the moving image, which is transformed (generally with the help of deformation techniques) to closely match another image called fixed image. For a single patient, this is often used to compare several imaging modalities (e.g., CT and MRI). For example, [6] by using deformable image registration technique between a healthy person and original patient, Krivov et al. [8] established a technique that allowed brain lesions to be projected onto scans of healthy patients. Similarly, in their demonstration of a method using diffeomorphic image registration, Nalepa et al. [9] co-registered pairs of lesion pictures to produce enhanced data that, when paired with an affine augmentation, increased the generalization of their Deep Learning models.

3.3 GAN-Based Augmentation Methods

As discussed, a brief introduction on GANs in previous section, detailed architecture will be discussed in subsequent section of this chapter. From the above discussion, it is known that GAN is a deep learning-based augmentation technique to generate the data. As shown in Fig. 5 [6], through GAN-based CT, a lung nodule image have been generated.

Image Segmentation Using GAN -Based Augmentation Methods Instead of creating simply synthetic images, as is the case for classification tasks, the aim of data augmentation for image segmentation tasks is to produce synthetic image-label pairs [10]. A label in this context is often an image whose pixels or voxels have been given a category index that represents a specific meaningful entity. As an illustration, consider creating artificial medical images using the semantic labeling of anatomical structures. A semantic label in this context is an image of a label on which each pixel or voxel has been given a class index. Figure 6 displays an illustration of how a CGAN can be used to create artificial MRI images using semantic labeling.

The labels and generated images can be combined to form the training set for a segmentation network. Cao et al. [11] and Shin et al. [12] both used this kind of label image translation technique to synthesize aberrant brain MRI pictures and PET and CT images, respectively, for tumor segmentation. Synthetic images with

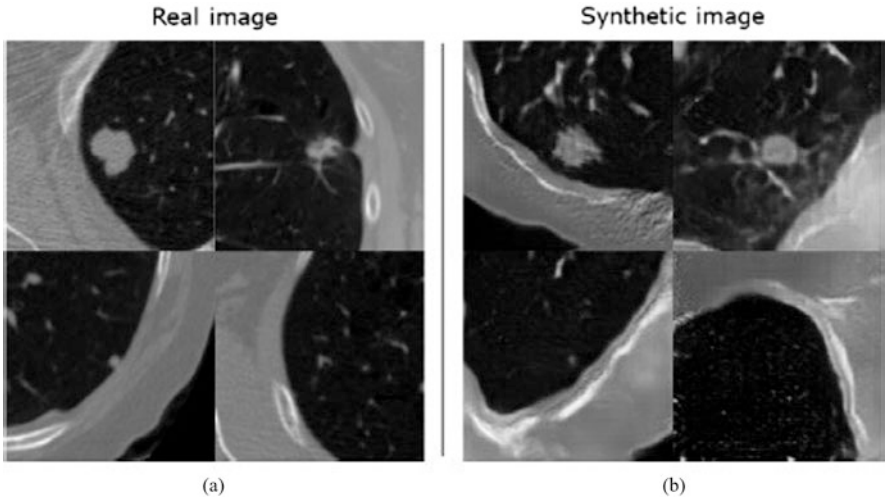


Fig. 5 (**a**) Real images of CT Courtesy: LIDC lung nodule dataset; (**b**) generated images using GAN [6]

diverse anatomical appearances were created using enhanced labeling, where the location, shape, and size of tumors can be altered. Jiang et al. [13] performed faux MR image synthesis from CT pictures under the constraints of two different GAN models simultaneously trained for segmentation of lung tumor in MRI. For the purpose of segmenting organs, Sandfort et al. [14] converted CT images that were enhanced in contrast to un-enhanced ones using CycleGAN.

These were few examples of works conducted earlier, which show the importance of data augmentation and GANs and its variants in medical stream. Further research and work are also going on in this direction. The next subsequent section focuses on the discussion on one of the important variants, that is, Cycle Consistent GAN, its architecture, working, and its use case in healthcare.

4 Cycle Consistent Generated Adversarial Networks Architecture

4.1 Introduction to Cycle Consistent GANs

Every GAN-based approach, including CycleGAN, inherently produces content that is hallucinatory in nature. Its outputs are scenarios of "what might it look like if..," and although reasonable, the forecasts may be very different from reality. As discussed above in the Introduction, GANs in general have a discriminator and a generator. Both discriminator and generator are complementary of each other. In

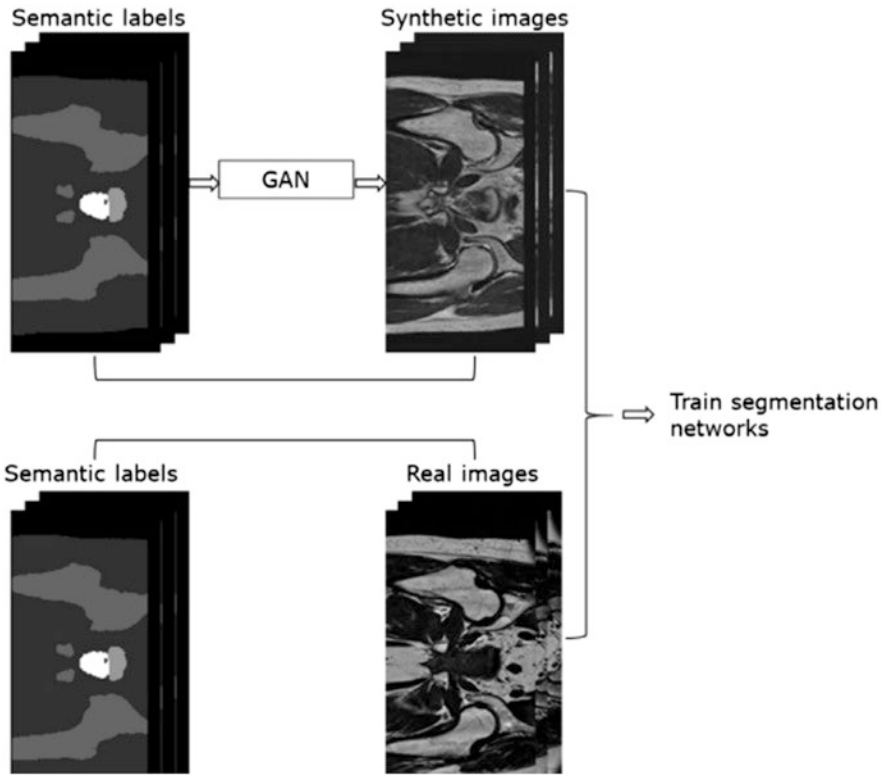


Fig. 6 The picture is taken from a prostate MRI dataset [10]

CycleGANs, there are two GANs, that is, there are two discriminators and two generators.

For an illustration, let us consider the same horse and zebra example as discussed in the Introduction above. One of the generators of CGAN will be transforming the image of a horse to a zebra and other transforming a zebra to a horse. Why is this transformation needed? We know that transformation is the basic key in GANs for generating new data. So, basically, CycleGAN is a paradigm for translating images to images. These are similar to the Pix2Pix model but in Pix2Pix the requirement is that the training data needs to be paired. However, it does not work for unpaired image dataset. CycleGANs are used for this purpose. Hence, using the CycleGAN method, image-to-image translation models can be trained irrespective of paired images. To do so it makes use of unsupervised learning for model training by utilizing a set of pictures from both source and target domains which are not related. CycleGAN architecture was proposed by Jan-Yan Zhu et al. [15] along with his associates in their work unpaired image-to-image *Translation Cycle Consistent Adversarial Networks*.

In the above example of the horse and zebra, the horse and zebra are unpaired images. Discriminators are used throughout the training phase to determine whether generated images are authentic or false. With the use of their respective discriminators' input, this approach can help generators improve [15]. A generator in CycleGAN receives additional input from the previous generator. This input or feedback makes sure that an image produced by a generator is cycle consistent, which means that using two generators in succession should provide an image that is similar. This can be understood by Fig. 7. Suppose zebra belongs to domain1 and horse belongs to domain 2, then the generator (GEN 2) corresponding to domain 2 is applied to this image to generate the image in domain 2. Similarly, images will be translated from domain 2 to domain 1 by applying GEN 1. Discriminators will be checking whether generated images are real or fake. Hence, the role of discriminator1 is to determine whether the outputs of Gen1 are true or false in terms of Domain 2, they are fed through discriminator 2. Similarly, role of discriminator2 is to determine whether the outputs of Gen2 are true or false in terms of Domain 1, they are fed through discriminator 1.

This is the basic for CycleGAN. Cycle consistent GANs implement an extension to it, that is, they bring in the concept of cycle consistency. In cycle consistent GAN, it is proposed that the output image from first generator is given to the second generator as input. The output from second generator must be almost similar to the actual image fed to the first generator. So, in cycle consistent GANs, the main aim is to learn the mappings Gen1 which maps the input image from domain1 to domain2 and Gen2 that maps an image from domain 2 to domain1 [15]. Also, the two adversarial discriminators can differentiate between real images and fake images. This introduces 2 important terms:

- Adversarial losses: to match the distribution of output images to the distribution of data in the target domain.
- Cycle consistency losses: to stop the learnt mappings Gen1 and Gen2 from conflicting with one another.
- In a simple way, if I_1 is an image belonging to a domain D1 and we apply the mapping (Gen1) on it, it produces an image in domain2 say, I_2. Now if we apply mapping (Gen2) on this generated image I_2, it will produce an image back in domain 1 say as I', then the cycle consistency loss is given as:

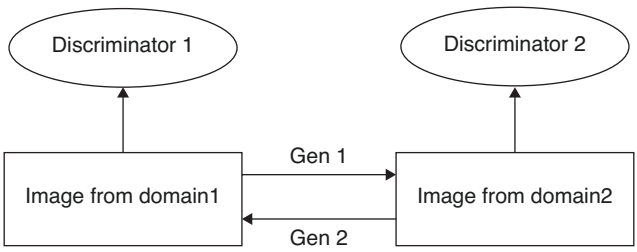


Fig. 7 CycleGAN process

$$\text{CCL} = I_2 - I_1 \tag{3}$$

The above equation is the logical way of looking at the cyclic loss. More detail of these two losses will be discussed later in this chapter.

5 Building Cycle Consistent GANs

As we have seen above, two important mappings have to be learnt, that is, generator and discriminator, and we will see how these mappings are developed in cycle consistent GANs [16].

5.1 Generator

The functioning of a generator can be divided into three main parts:

- *Encoder*: It is meant to extract the features from the image. The first phase involves using a convolution network to separate the features from an image. A convolution network takes an image as its input, the filter window size which is scanned to extract image features, and the size of the stride to determine by what amount to move the filter window after each advancement. Each convolution layer causes the extraction of dynamically higher level features.
- *Transformer*: These convolution layers can be considered as linking various nearby image features, and based on these features, decisions can be made about how we would want to alter the element encoding of an image starting with one and moving on to the next. A transfer learning model can be used for that. Suppose we use six to nine layers of Resnet blocks. Two convolution layers make up the Resnet block where more information is added to the output. As a result, the qualities of unique images will not be maintained in the yield and the results will be unexpected. This is done to ensure that properties of the contribution of previous layers are accessible for subsequent layers and also to ensure that their yield does not deviate significantly from unique information.
- *Decoder*: Output generated by transformer is passed to the decoder. To restore the representation's original size, the decoder utilizes a 2-deconvolution block of fraction strides. So, basically it is performing the inverse of what was done by the encoder.

The above process can be represented as shown in Fig. 8.

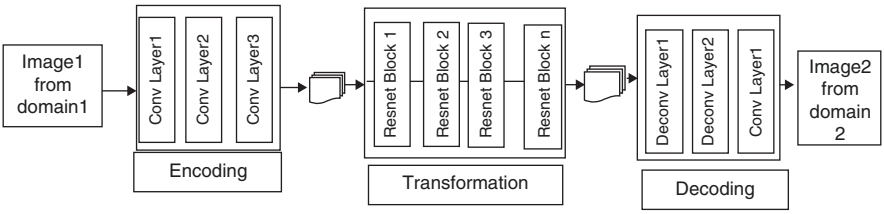


Fig. 8 Generator architecture

5.2 Discriminator

- The discriminator would use images to attempt to determine if they were authentic or false. It works along with a transformer. It will take the input image and extract features from it. It checks whether these features are as per classification or not. It can be represented as shown in Fig. 9.

5.3 The Loss Function

As discussed in the previous section, cycle consistent GANs suffer from 2 types of losses, that is, cycle consistent loss and adversarial loss. Let us have a deeper insight into it. The loss function must be designed in such a way that cycle consistent GAN is able to achieve its aim. The model has 2 generators and 2 discriminators. Loss can be viewed at the following points:

1. All the images belonging to all categories must be accepted by the discriminator.
2. The discriminator should try to reject all those images which the generator is trying to pass as valid images.
3. The generator must try that the discriminator passes all the generated images as valid images.
4. The generator creates a fake image using Image1 → Image2, then we should almost surely return to a unique picture using another Generator Image2 Image1 → – it must satisfy cyclic consistency. The created image thus returns the original image.

The Objective Function The cycle consistent GAN, therefore, based on the above four points, can have two components: cycle consistent loss and adversarial loss (as also discussed above) [17]. Consider two generators in the models as U and V, two discriminators as D_a and D_b, and the domains A and B to which the images belong. Suppose there are n number of images in the dataset. Consider U is trying to translate A into outputs, which are fed through D_a to check whether they are real or fake according to Domain B. Similarly, V is trying to translate B into outputs, which are

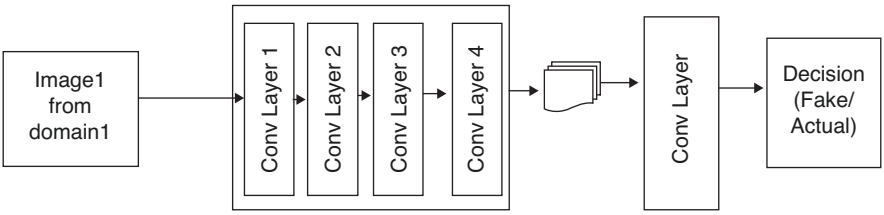


Fig. 9 Discriminator architecture

fed through D_b to check whether they are real or fake according to Domain A. Adversarial Loss based on the above can be given as follows:

$$\text{Loss}_{AD}\left(U,D_b,A\right)=1/n\sum_{i=1}^{n}\left(1-D_b\left(U\left(a_i\right)\right)\right)^2 \tag{4}$$

$$\text{Loss}_{AD}\left(U,D_a,B\right)=1/n\sum_{i=1}^{n}\left(1-D_a\left(V\left(b_i\right)\right)\right)^2 \tag{5}$$

Another component, that is, cycle consistency loss can be given as follows:

$$\text{Loss}_{\text{Cyc}}\left(U,V,A,B\right)=\frac{1}{n}\sum_{1}^{n}\left[V\left(U\left(a_i\right)\right)-a_i\right]+\left[U\left(V\left(a_i\right)\right)-b_i\right] \tag{6}$$

The above equation is similar to the logical Eq. (3) given above. Based on Eqs. (4), (5), and (6), the overall objective function can be defined as follows:

$$\text{Loss}_{\text{Tot}}=\text{Loss}_{\text{AD}}+\alpha\text{Loss}_{\text{Cyc}} \tag{7}$$

where α is weighting hyper-parameter for cycle consistency loss and is recommended to be set to a value of 10. However, it can be fine-tuned as well. In order to update weights and biases so that error can be reduced, an optimizer like ADAM, stochastic gradient descent, Adagrad, etc., can be used. The discriminators can be fed with previously generated images rather than just those created by the most recent versions of the generator in order to prevent the model from altering much from iteration to iteration. This is usually done if the model gets stuck in local oscillations.

6 Applications of Cycle Consistent GANs

There are various domains where data augmentation is required and where various other models and other variants of GAN fail because either data is unlabeled or belongs to unpaired images. In such cases, Cycle Consistent GANs have proved to be very useful. Some of the applications in brief are given here:

(a) *Style Transfer*: The technique of transferring a particular aesthetic to another domain, usually photography, is known as style transfer. Using the aesthetics of Monet, Van Gogh, Cezanne, and Ukiyo-e to landscape pictures serves as an illustration of the CycleGAN [16].
(b) *Object Transfiguration*: It is the transition of an object from one class into another, for example, from a dog to a cat. It is shown how the CycleGAN may turn pictures of horses into pictures of zebras and the other way around. Given that horses and zebras are comparable in size and structure, with the exception of color, this kind of transformation seems plausible [18].
(c) *Season Transfer*: In this process photos taken in one season like summer are converted to another season, like winter. On images of winter landscapes converted to summer landscapes, and vice versa, the CycleGAN is used to illustrate its capabilities [19].
(d) *Photographs Generated from Paintings*: As the name implies, photograph creation from paintings is the synthesis of photorealistic photographs given a painting, usually by a well-known artist or renowned setting. The CycleGAN is used to convert several Monet paintings into credible pictures [20].
(e) *Photograph Enhancement*: The term "photographic enhancement" describes changes that make the original image better in some way. By enhancing the depth of field (e.g., by adding a macro effect) on up-close photos of flowers, the CycleGAN is used to show photo enhancement [21]. Other than these small applications cycle consistent GANs have been used extensively in enhancing images in the medical field and to augment data in medical sciences. The next section of the chapter focuses on various applications of cycle consistent GAN in Healthcare.

7 Cycle Consistent GAN in Healthcare

There are various data augmentation techniques which have been applied so far as discussed in the previous section of the chapter. As seen before, GANs and their variants have been successfully applied in medical science field. Here, the main focus of the study is how cycle consistent GANs have been used in healthcare. In Morís et al. [22], the authors have proposed the use of cycle consistent GANs for the study of Novel Coronavirus which was related to the recent pandemic situation. This disease primarily affects the patient's respiratory system and can cause pneumonia and severe acute respiratory syndrome instances, which cause the

development of many abnormal structures in the lungs. Chest X-ray imaging can be used to study these diseased structures. The usage of portable chest X-ray equipment rather than traditional permanent gear is advised for the health services to stop the spread of the virus. Yet, there are various issues with portable devices (especially those related with capture quality). Hence, with respect to low-quality and less informative datasets collected through portable devices, Morís et al. [22] suggested that multiple cycle generative adversarial networks be used to create interesting and pertinent synthetic images in order to improve the effectiveness of COVID-19 screening in order to overcome the dearth of COVID-19 samples. Without the need for paired data, they used three complimentary CycleGAN designs to simultaneously produce a fresh collection of artificial portable chest X-ray pictures from three distinct circumstances (normal, diseased, and genuine COVID-19). In addition, four different CycleGAN variants (Unet-128, Unet- 256, ResNet-6, and ResNet-9) were tested for every scenario for proper and extensive validation of methodology, evaluating 12 different approaches. Then, in order to add more dimensions to the original dataset and improve the training for the COVID-19 screening procedure, this new collection of artificial images is introduced to the dataset. They used CHUAC dataset where they retrieved 720 images of different resolutions. The dataset had 240 COVID-19 data points, 240 data points belonging to patients with no COVID-19 but having other lung-related issues having symptoms close to COVID-19, and 240 data objects from healthy volunteers as well as patients with other pathological conditions that are unrelated to the symptoms of COVID-19 disease. All the images were unpaired. Using this unpaired dataset, learning of correlation was done by CycleGAN between the input and output images for producing a new collection of chest X-ray images. Moreover, the architecture's cyclic nature enables a reverse transformation, meaning the model may turn a generated chest X-ray image back into the original chest X-ray image. The main workflow they have used here is shown in Fig. 10 [22]. Validation accuracies were 95.83%, 97.92%, 100.00%, and 96.88% for Unet-128, Unet-256, ResNet-6, and ResNet-9, respectively.

In another approach presented in Nalepa et al. [23], authors presented the application of cycle consistent GAN to augment the data for brain tumor segmentation. In this work, the authors used CycleGAN to perform adaptation of domain to deal with the various data distributions (from synthetically produced phantom data to the actual BraTS MRI scans). Findings from experiments demonstrated that the domain adaptation could produce images that were virtually identical to the original data and could thus be included in the training set without risk.

In a recently released article [24], investigators classified breast cancer data for a pathological healthcare job using a Cyclic GAN model. Concerns about inconsistent staining with different batches of pathology images are frequent. Using these samples to train the classification model decreases the classification accuracy. Rich textural features and minimal semantic information can both be found in pathological images. Further medium- and low-level variables must be retrieved to improve classification accuracy. Their research offered a method that used CycleGAN and an improved DPN (dual Path Network), as well as a color normalization technique based on CycleGAN, to overcome the afore- mentioned problems and lessen the

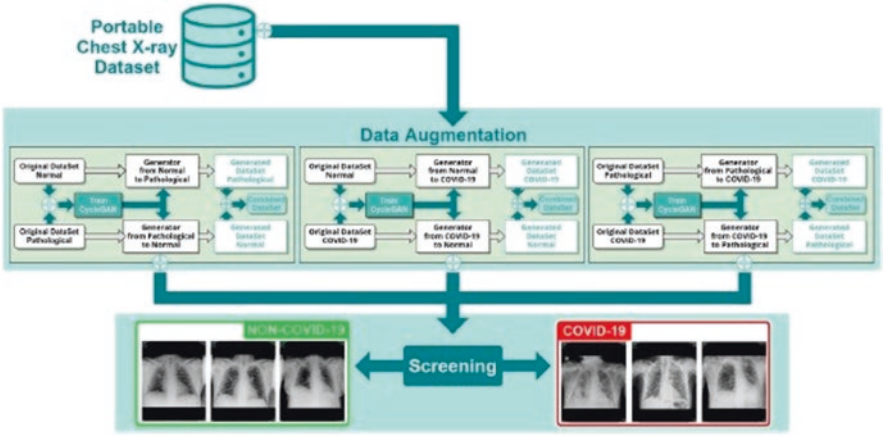


Fig. 10 Main workflow for COVID-19 study using CycleGAN

impact of dyeing concerns on classification accuracy. Cycle consistency enables the CycleGAN to provide images that are more precise and reliable. They got and advantage of using Cyclic GAN that it was faster in operation than CNN. Another advantage was that it did not need as much pre-processing, but it still had time and space complexity issues like CNNs and RNNs. In order to lessen the influence of color on classification, diseased images with varied colors were also converted to the same color using the CycleGAN. For this work, they used the BreaKHis breast cancer pathological image data collection, which includes 7909 tagged breast cancer pathological images from 82 breast cancer patients. The accuracy rate for the CycleGAN model's data categorization is 10% higher than it would be without the normalization strategy, and the false detection rate and missed detection rate are all reduced. They also showed that CycleGAN-based color normalization for the diseased images described in this study is effective and reduced the effect of uneven staining on the classification of pathological images.

For the automatic detection of COVID-19 using X-ray images, CycleGAN and transfer learning algorithms have been presented in Bargshady et al. [25]. For validation of their model, they used the Extensive COVID-19 X-ray and CT Chest Images Dataset. However, for training purpose they used same dataset with images augmented using cycle consistent GANs. By converting normal images into COVID-19 images and COVID-19 photos into normal images, CycleGAN had been utilized to produce and supplement data. In order to classify chest X-ray images with or without COVID-19 properties, the CycleGAN-Inception model was created. Examples of the X-ray pictures produced by CycleGAN – (a) a selection of authentic and artificial images ranging from class non-COVID-19 to COVID-19; (b) genuine and created images from class COVID-19 to non-COVID-19 – are displayed as an example in Fig. 11 [25]. The collected findings of the work showed that the proposed CycleGAN-Inception was successfully able to distinguish between

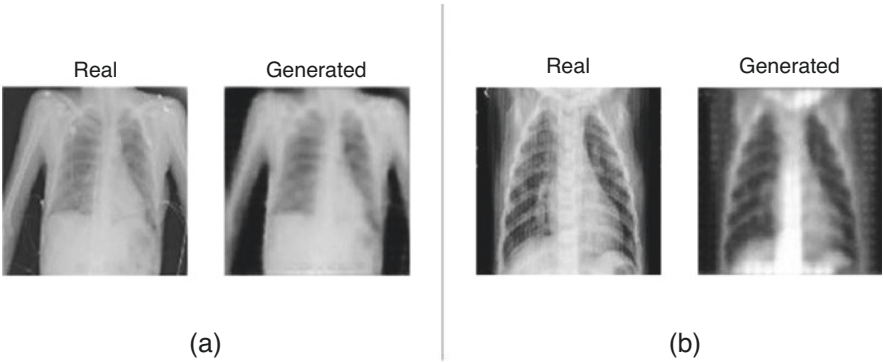


Fig. 11 Chest X-ray images generated through CycleGAN from (**a**) class non-COVID-19 to COVID-19 (**b**) from COVID-19 to non-COVID-19

COVID-19 and non-COVID-19 patients using radiographic pictures and hence can detect positive COVID-19 cases with high accuracy.

Another work was carried out by Harms et al. [26] where a paired cycle-GAN-based image correction method was proposed for quantitative cone-beam computed tomography (CBCT). Although CBCT enables routine 3D imaging, the practical application of CBCT is constrained by the pictures' severe artefacts. To learn a mapping between CBCT pictures and matched planning CT images, the suggested method incorporated a residual block notion into a cycle consistent adversarial network (CycleGAN) framework, known as res-cycle GAN. End-to-end CBCT-to-CT conversions are made possible by the generator's use of a fully convolution neural network with residual blocks. Twenty patient datasets for the pelvis and 20 patient datasets for the brain were used to evaluate the suggested technique. Using this technique improvements of 45%, 1%, 93%, and 16% in the brain, and 71%, 2%, 65%, and 38% in the pelvis, over the CBCT image were found for Mean Absolute Error (MAE), spatial non-uniformity (SNU), normalized cross-correlation (NCC) indices, and peak signal-to-noise ratio (PSNR).

In Yoo et al. [27], authors have done a feasibility study in order to enhance deep learning in optical coherence tomography (OCT) diagnosis of retinal diseases that are rare with few-shot classification. Because there is a dearth of training data for uncommon retinal illnesses, our strategy was FSL (Few Shot Learning) with data augmentation using GAN. To create images without matching paired images, the cycle consistent GAN (CycleGAN) was used. Supervised GAN approaches, such as conditional GAN and Pix2Pix, were not relevant in this study since there is no database that contains both abnormal OCT pictures and matching normal OCT images. Hence, authors used CycleGAN framework for this study. Each CycleGAN model was developed using two domains for training: the normal retina and a single uncommon illness. Both linear and elastic adjustments were used to improve the few-shot OCT pictures of uncommon disorders. Random rotation from −30° to +30°, zooming from 0% to 20%, left and right flip, width and height translation from 5% to +5%, and random brightness change from 10% to +10% were all examples of linear transformation. A Gaussian kernel was used to produce elastic

transformation. 2000 normal retinal OCT images from Kermany's [28] work were randomly selected for this training step, while 2000 diseased images were created using few-shot basic augmentation. The five trained CycleGAN models translated the pathological images associated with each uncommon disease from the normal OCT images. Ophthalmologists with extensive training examined the generated images and eliminated any with obvious artefacts. To train the diagnostic classifier model, a total of 5000 pathological OCT images – 3000 CycleGAN-based and 2000 basic enhanced images – were created for each rare disease. Authors have also built their segmentation model using CycleGAN. In Mohamadipanah et al. [29], another recent work, authors have used CycleGANs for generating rare surgical events. Surgery diseases ranging from benign to malignant illnesses are treated by pulmonary lobectomy. Here, CycleGAN was trained using six videos of minimally invasive lobectomies, 1819 of which included no bleeding and 3178 of which contained significant bleeding. On a fresh video that was not used during training, the CycleGAN algorithm's performance was evaluated. The trained CycleGAN was able to transform the laparoscopic lobectomy photos in accordance with the massive bleeding images that were matching to them, while preserving the original images' content (such as the location of instruments in the scene) and changing each image's style to massive bleeding (i.e., blood artificially introduced to selected places on the images). CycleGANs have shown good improvements here.

8 Future Scope and Conclusion

In a similar way as discussed in the previous section, various other research works have been carried out in healthcare industry using CycleGAN framework to generate more data and help the research and disease diagnosis and improvements. Radio imaging, CT scans, MRI images, etc., in all of these techniques improvements have been achieved. It also has proved to be an effective technique for histopathology image generation. In the previous section, 2 rare medical events have also been discussed where CycleGANs have shown fruitful results. There are many more such rare diseases for which research is still going on. CycleGANs can be an answer, along with deep learning, to such events where due to data insufficiency researchers are unable to come with proper treatments or medicines. We have diseases like osteoporosis, hypophosphatasia, and black bone disease that cause osteoarthritis and cystic fibrosis, to name a few, for which still we have no cure. Data for such disease is not available in adequate amount. Also certain times viruses are unpredictable like HIV. Augmentation techniques especially unsupervised like CycleGAN can be an answer through which scientists, researchers, and doctors can bring a huge change. The combination of CycleGANs with other deep learning methods can be used for multiple such problems in various domains. New models can be generated using CycleGAN and different deep learning approaches to produce further better results and to find answers for various unanswered problems in the medical area.

References

1. Foster, D. (2022). *Generative deep learning*. O'Reilly Media, Inc.
2. Jakub, L. (2019). *Bok Vladimir: GANs in action: Deep learning with generative adversarial networks*. Manning Publication.
3. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), 1–48. https://doi.org/10.1186/s40537-019-0197-0
4. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618–626).
5. Goel, K., Gu, A., Li, Y., & Ré, C. (2020). Model patching: Closing the subgroup performance gap with data augmentation. In *International conference on learning representation*. https://openreview.net/forum?id=9YlaeLfuhJF
6. Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology, 65*(5), 545–563.
7. Widianti, L. W., Sudiro, S. A., Madenda, S., & Harlan, J. (2020). Cropping method on gray-scale images for periapical radiographs of human teeth. In *IOP conference series: Materials science and engineering* (Vol. 879, No. 1, p. 012114). IOP Publishing.
8. Krivov, E., Pisov, M., & Belyaev, M.. (2019). MRI augmentation via elastic registration for brain lesions segmentation. Polyp detections in CT colonography. In *Medical imaging 2019: Computer-aided diagnosis*.
9. Nalepa, J., Marcinkiewicz, M., & Kawulok, M. (2019). Data augmentation for brain-tumor segmentation: A review. *Frontiers in Computational Neuroscience, 13*, 83. https://doi.org/10.3389/fn-com.2019.00083
10. Dowling, J. A., Sun, J., Pichler, P., et al. (2015). Automatic substitute computed tomography generation and contouring for magnetic resonance imaging (MRI)-alone external beam radiation therapy from standard MRI sequences. *International Journal of Radiation Oncology, Biology, Physics, 93*, 1144–1153.
11. Cao, K., Bi, L., Feng, D., & Kim, J. (2020). Improving PET-CT image segmentation via deep multi-modality data augmentation. In *Machine learning for medical image reconstruction*. Lecture Notes in Computer Science (pp. 145–52).
12. Shin, H. -C, Tenenholtz, N. A., & Rogers, J. K., et al. (2018). Medical image synthesis for data Aug- mentation and anonymization using generative adversarial networks. In *Simulation and synthesis in medical imaging*. Lecture Notes in Computer Science (pp. 1–11).
13. Jiang, J., Hu, Y.-C., Tyagi, N., et al. (2019). Cross-modality (CT-MRI) prior augmented deep learning for robust lung tumor segmentation from small MR datasets. *Medical Physics, 46*, 4392–4404.
14. Sandfort, V., Yan, K., Pickhardt, P. J., & Summers, R. M. (2019). Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific Reports, 9*, 16884.
15. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).
16. https://github.com/junyanz/CycleGAN
17. Tripathy, S., Kannala, J., & Rahtu, E. (2019). Learning image-to-image translation using paired and unpaired training samples. In C. Jawahar, H. Li, G. Mori, K. Schindler (Eds.), *Computer vision – ACCV 2018*. Lecture Notes in Computer Science (vol. 11362). Springer.
18. Chen, X., Xu, C., Yang, X., & Tao, D. (2018). Attention-GAN for object transfiguration in wild images. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 164–180).

19. Paudel, B. H., & Sah, R. K. (2021). Landscape image season transfer using Generative Adversarial Networks. In *Proceedings of 10th IOE graduate conference* (Vol. 10, pp. 2350–8906).
20. Bellale, V., Kashyap, S. K., Rawat, V., Shinde, N., & Kale, R. (2022). Artistic style generation using cycle GAN. In *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)* (Vol. 2, Issue 3).
21. You, Q., Wan, C., Sun, J., Shen, J., Ye, H., & Yu, Q. (2019). Fundus image enhancement method based on CycleGAN. In *2019 41st annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 4500–4503). Berlin, Germany. doi: https://doi.org/10.1109/EMBC.2019.8856950.
22. Morís, D. I., de Moura Ramos, J. J., Buján, J. N., & Hortas, M. O. (2021). Data augmentation approaches using cycle-consistent adversarial networks for improving COVID-19 screening in portable chest X-ray images. *Expert Systems with Applications, 185*, 115681. https://doi.org/10.1016/j.eswa.2021.115681
23. Nalepa, J., Marcinkiewicz, M., & Kawulok, M. (2019). Data augmentation for brain-tumor segmentation: A review. *Frontiers in Computational Neuroscience, 13*, 83. https://doi.org/10.3389/fncom.2019.00083
24. Chopra, P., Junath, N., Singh, S. K., Khan, S., Sugumar, R., & Bhowmick, M. (2022). Cyclic GAN model to classify breast cancer data for pathological healthcare task. *BioMed Research International, 2022*, 6336700. https://doi.org/10.1155/2022/6336700
25. Bargshady, G., Zhou, X., Barua, P. D., Gururajan, R., Li, Y., & Acharya, U. R. (2022). Application of CycleGAN and transfer learning techniques for automated detection of COVID-19 using X-ray images. *Pattern Recognition Letters, 153*, 67–74. https://doi.org/10.1016/j.patrec.2021.11.020
26. Harms, J., Lei, Y., Wang, T., Zhang, R., Zhou, J., Tang, X., et al. (2019). Paired cycle-GAN-based image correction for quantitative cone-beam computed tomography. *Medical Physics, 46*(9), 3998–4009. https://doi.org/10.1002/mp.13656
27. Yoo, T. K., Choi, J. Y., & Kim, H. K. (2021). Feasibility study to improve deep learning in OCT diagnosis of rare retinal diseases with few-shot classification. *Medical & Biological Engineering & Computing, 59*, 401–415. https://doi.org/10.1007/s11517-021-02321-1
28. Kermany, D. S., Goldbaum, M., Cai, W., et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell, 172*, 1122–1131.e9. https://doi.org/10.1016/j.cell.2018.02.010
29. Mohamadipanah, H., Kearse, L., Wise, B., Backhus, L., & Pugh, C. (2023). Generating rare surgical events using CycleGAN: Addressing lack of data for artificial intelligence event recognition. *Journal of Surgical Research, 283*, 594–605. https://doi.org/10.1016/j.jss.2022.11.008

Geometric Transformations-Based Medical Image Augmentation

S. Kalaivani, N. Asha, and A. Gayathri

1 Introduction

Deep convolutional neural networks achieved extraordinarily fit on numerous computer vision tasks. Nevertheless, the networks of such models are seriously dependent on huge data to evade overfitting. Overfitting takes place while the model attempts to cover all the data points or further than the essential data points existing in the specified dataset. The above causes the network to begin accumulating the dataset's noise and incorrect values, which lowers both the model's accuracy and effectiveness. State-of-art provides a better solution in this area and generates more datasets in medical images. The majority of survey participants recommended data augmentation as a data-space solution to the issue of limited data. In order to improve deep learning models, a variety of strategies are referred to as "data augmentation." These techniques aim to increase the quantity and quality of training datasets. Geometric transformations, color space augmentations, random erasing, kernel filters, feature space augmentation, mixing images, generative adversarial networks, meta-learning, adversarial training, and neural style transfer are some image augmentation algorithms. In discriminative tasks, deep learning models have made great strides. This is made possible by the development of deep network architectures, the accessibility of vast quantities of data, and sophisticated processing. For the growth of convolutional neural networks (CNNs), deep neural networks are effectually used for computer vision applications like image analysis, object recognition, and image segmentation. The neural network preserves the spatial properties of images. The convolutional layers fetch the depth of the image to increase the feature maps through successive down-sampling [1–5]. The popularity

S. Kalaivani (✉) · N. Asha · A. Gayathri
School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamil Nadu, India
e-mail: kalaivanis@vit.ac.in; nasha@vit.ac.in; gayathri.a@vit.ac.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_8

of CNNs has increased passion and hope for the successful application of deep learning to computer vision problems.

There exist numerous academic fields that use Deep Convolutional Networks to tackle computer vision problems in an effort to outperform current benchmarks. One of the hardest problems is enhancing these models' capacity for generalization. A model's performance on data it has previously seen (training data) versus data it has never before seen is measured by its generalizability (testing data). The model's poor generalizability is identified by plotting the validation accuracy and training set in every epoch as shown in Fig. 1. The graphic in Fig. 1a demonstrates an inflection point when the training efficiency keeps decreasing while the error value starts to rise. As a result of the model being overfit to the training data as a result of the increased training, the model did better on the testing set than the training sample. In Fig. 1b, in comparison, a model with the intended correlation among training and testing error is depicted in the plot at the bottom.

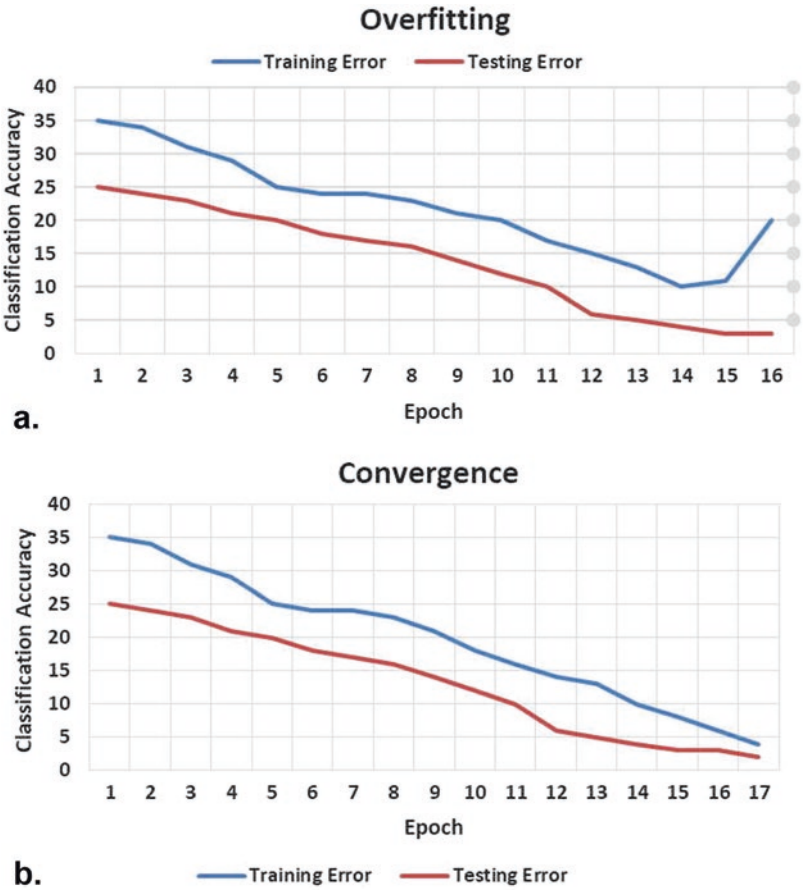


Fig. 1 (**a**) Overfitting (**b**) Convergence

The validation error will decrease in tandem with the training error in order to develop practical Deep Learning models. Data enhancement is an effective way to accomplish this. By providing a wider variety of potential data points, the augmented data will narrow the gap between the sets used for validation and training as well as any subsequent testing sets. It is useful to set the scene and take into account what makes image identification such a challenging process in the first place before considering image augmentation techniques. The image recognition methods must overwhelm problems with, lighting, viewpoint, background, scale, occlusion, and other factors in traditional discriminative. These challenges are addressed by data augmentation, which aims to incorporate inherent translational invariances within the dataset to improve the performance of the final models. The idea that larger datasets produce stronger Deep Learning models is well acknowledged [6, 7]. Yet, given the labor-intensive nature of data collection and labelling, compiling huge datasets can be a highly challenging task. The problem of small datasets is one that medical image analysis faces frequently. Esteva et al. [8], have proven that medical image processing with deep convolutional networks is extremely effective for skin lesion categorization tasks for given large datasets. This has motivated the application of CNNs to a variety of medical image analysis applications, including the categorization of skin lesions, brain scan analysis, liver lesions, and more [9]. The majority of the images examined come from expensive and time-consuming scans like magnetic resonance imaging (MRI) and computed tomography (CT). The rare nature of diseases, the confidentiality of patients, the need for medical specialists for categorizing, the cost, and the amount of labor needed to perform medical imaging processes make it particularly challenging to generate large medical picture collections. Numerous works on image data augmentation, in particular GAN-based oversampling, have been inspired by the use of medical image categorization.

LeNet-5 [10] contains a data warping technique for image enhancement. One of the first CNN applications for classifying handwritten digits was this one. Applications for oversampling have also looked at data augmentation. The model will not be overly biased toward categorizing samples as belonging to the dominant class type if uneven class distributions are resampled by oversampling. Images from the minority class are randomly copied until the required class ratio is obtained via a simple technique called random oversampling (ROS). Chawla et al. [6] developed SMOTE (Synthetic Minority Over-sampling Technique), which is the predecessor to intelligent oversampling methods. By k-Nearest Neighbours, SMOTE and the expansion of Borderline-SMOTE [8] interpolate new points using existing instances to construct new instances. SMOTE was mostly used for tabular and vector data, and its major objective was to address problems caused by class inequality.

The system of classification of chest X-ray anomalies, the detection of lung cancer, the generation of high-quality skin lesions, and the synthesis of brain MRI are all included in the exploration of the application of GAN image synthesis in medical imaging applications. The use of GANs in rebuilding includes accelerated magnetic resonance imaging, CT de-noising, and PET de-noising, as well as utilization of super-resolution GANs in the retinal vasculature. In order to classify liver lesions, Frid-Adar et al. [11] applied GAN-based image synthesis data augmentation which

increases the performance of classification from 78.6% sensitivity and 88.4% specificity by means of 92.4% specificity and traditional augmentations to 85.7% sensitivity [12].

2 Geometric Transformations: Basic Manipulation

To expand the size of the training dataset, geometric transformation involves changing the original image in a variety of ways, including translation, scaling, rotation, flipping, and resizing [13]. These traditional methods of data augmentation provide relatively related images [14] and severely decrease the model's ability to learn from and generalize the test data.

Flipping Flipping the image pixels vertically or horizontally as per the necessity. Horizontal flipping is the most popular type of flipping since it is more realistic. For instance, a dataset comparing cats and dogs might contain all the photos of cats facing the spectator's left. Unsurprisingly, dogs moving to the right may be misclassified by the trained model. The best method to solve this issue is to amass more training photos with as many distinct points of view as you can. One of the most simple methods to broaden the scope or diversity of data is flipping. When the data has special qualities, though, it might not be appropriate. Asymmetric or direction-sensitive data, such as letters or digit numbers, cannot employ the flipping technique because it produces wrong labels, or even opposite labels, according to the concept of label safety explained in Shorten et al. [12].

Rotation Rotating the image by an angle ranging from 0 to 360 degrees as per the necessity, the degree is varied. A straightforward geometric data augmentation method is rotation. The photographs are rotated by a predetermined angle and utilized as training examples alongside the original images. Rotation has the drawback of potentially causing information loss at the image boundary. The border issue of the rotated pictures can be resolved in a number of ways, including random closest neighbor rotation (RNR), random reflect rotation (RRR), and random wrap rotation (RWR). Specifically, the RNR methodology fills in the black areas by repeating the nearest pixel values, while the RRR technique uses a mirror-based method and the RWR technique makes use of the periodic boundary strategy.

Translation Shifting the axes of the images as per the necessity.

Scaling Varying the dimension either through cropping the image (decrease the size) or enlarging the image (increase the size) as per the necessity.

Cropping The fundamental augmentation technique of cropping involves selecting a random portion of the target image and then scaling that portion back to the original size. Cropping images to a specific size before training is a common practice since training data may contain examples of varying sizes [12]. It is important to

note that cropping could result in samples with false labels. When using the cropping technique, for instance, images with multiple objects that are labelled according to the object with the dominant size may encounter a problem. In this situation, it is feasible to crop the image to show more of the supporting object rather than the dominant object.

The training data which contains the positional biases are addressed through geometric transformation. In fact, many sound sources of bias might distinguish the dissemination of the training data from the testing data. Geometric transformations are helpful not just because they are effective at overcoming positional biases but also because they are simple to employ. The use of operations like rotation and horizontal flipping is made simple by the abundance of imaging processing packages. Geometric transformations have several drawbacks, such as requiring more memory, costing more to compute, and requiring more training time. To ensure that some geometric alterations, such as translation or random cropping, did not change the image's name, they must be manually checked. As a result, there are relatively few situations in which geometric transformations can be used.

3 Test-Time Augmentation (TTA)

Researchers' interest in an innovative image augmentation technology has grown over the past few years. The scientific community was given a mathematical definition of TTA by Wang et al. [15]. They frame TTA as an inference issue with prior distributions and hidden variables. The images result in the hidden parameters of the elaboration process are thus seen as the end product of the process of production with concealed parameters. The evaluation of structure-wise uncertainty brought on by noise and picture modifications is the ultimate goal. In addition to the methods already stated, TTA produces a number of augmented images of the test set, inputs these augmented images to the training set, and then outputs an ensemble of such predictions as an aggressive response [16]. Figure 2 shows the method of both train

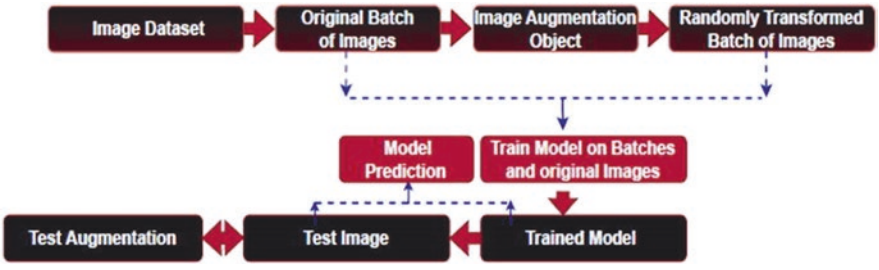


Fig. 2 Train and test time augmentation method

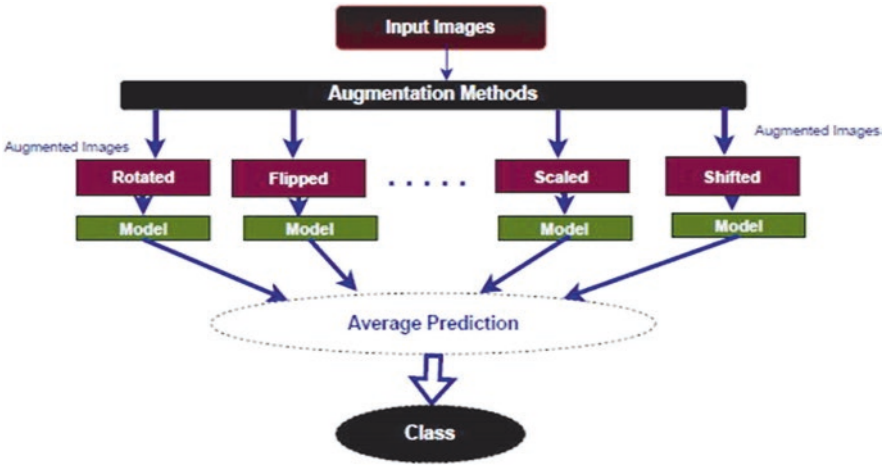


Fig. 3 Test-time augmentation framework

and test time augmentation, and Fig. 3. shows the test-time data augmentation (TTA) framework.

By assessing network stability and strength as real-world problems, TTA has opened up new possibilities for the medical imaging industry. In the instance of classification from mammograms, TTA can be utilized for algorithms that alter an input instance using affine, pixel-level, or elastic transformations. The academic community has concentrated on training data augmentations, but there is still plenty to learn about data transformation prior inference. To categorize a single image, TTA integrates several inference conclusions using various data augmentations.

4 Synchronous Medical Image Augmentation (SMIA) Framework

Chen et al. [17] proposed two methods based on synthesis and stochastic transformation. In the transform-based SMIA module, a subgroup of SMIA factors with such a random number of variables and randomized parameter values is chosen for every medical testing image and its tissue segments so as to simultaneously produce enhanced samples and the associated tissue segments. In the synthesis-based SMIA module, use an equal replacement method to synthesize new medical pictures while maintaining the original medical implications by replacing the original tissues at random with the enhanced tissues (Fig. 4).

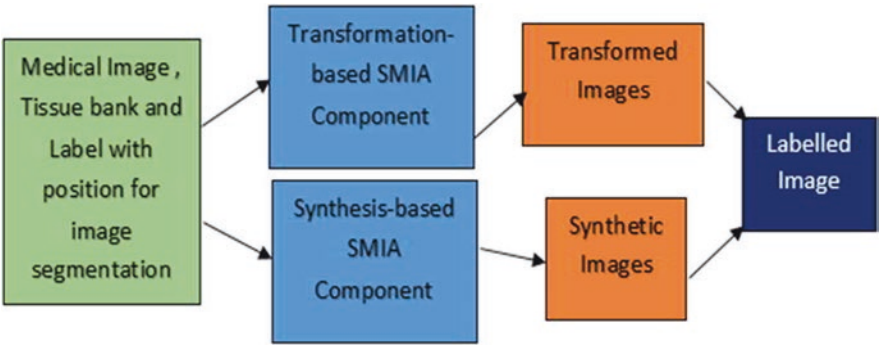


Fig. 4 Synchronous medical image augmentation (SMIA) framework

5 Random Local Rotation

Alomar et al. [18] introduces a novel data augmentation method in this part that falls under the typical data augmentation umbrella. It was influenced by methods that concentrated on particular regions of an image, such as the random erasing method.

Let D_i represent the practice data. A circular area of an image $I_i \, \varepsilon \, D_i$, with center coordinate *(xi, yi)* and radius *ri.* Let $d\theta \, \varepsilon \, [0, 2\pi]$ be the angle in variance in rotation. The first step is for every image I_i from the dataset D_i, select the circular area $C_{xi,yi,ri}$ from each image I_i with a randomly generated center *(xi, yi)* and radius *ri.*

The content within the circular area $C_{xi,yi,ri}$ is rotated to an angle $d\theta$, and added with to the outer region which is kept constant and is treated as newly generated image I_n. Lastly, image I_n is used to augment the original training dataset I_i. Two ways to add to the dataset are suggested. One way to replace the generated image is to the original image in the original dataset where the dataset size is unchanged only the data got changed. The other way is to add the generated original image to the original dataset.

6 Conclusion

Geometric transformation-based augmentation techniques are basic techniques where most of the other techniques are integrated with this during the pre-processing steps. Most of the deep learning techniques need more sample for training and it is hard to collect in real-time. So all these techniques are efficient to create more samples from the original image and models are converged with the training and testing samples. In this chapter geometric-based augmentation techniques are discussed in more detail.

References

1. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005
2. Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (pp. 843–852).
3. Wang, Y., Yu, B., Wang, L., Zu, C., Lalush, D. S., Lin, W., Wu, X., Zhou, J., Shen, D., & Zhou, L. (2018). 3D conditional generative adversarial networks for high-quality PET image estimation at low dose. *NeuroImage, 174*, 550–562. https://doi.org/10.1016/j.neuroimage.2018.03.045
4. Wolterink, J. M., Leiner, T., Viergever, M. A., & Isgum, I. (2017). 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
5. Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical Image Analysis, 58*, 101552. https://doi.org/10.1016/j.media.2019.101552
6. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*(16), 321–357. https://doi.org/10.1613/jair.953
7. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature, 542*(7639), 115–118. https://doi.org/10.1038/nature21056
8. Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proceedings of ICIC*. Lecture Notes in Computer Science (vol. 3644, pp. 878–87). 31. Ian.
9. Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems, 24*(2), 8–12. https://doi.org/10.1109/mis.2009.36
10. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791
11. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing, 321*, 321–331. https://doi.org/10.1016/j.neucom.2018.09.013
12. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1). https://doi.org/10.1186/s40537-019-0197-0
13. Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology, 65*(5), 545–563. https://doi.org/10.1111/1754-9485.13261
14. Shin, H. C., Tenenholtz, N. A., Rogers, J. K., Schwarz, C. G., Senjem, M. L., Gunter, J. L., … & Michalski, M. (2018). Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *Proceedings of the international workshop on simulation and synthesis in medical imaging.*
15. Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S., & Vercauteren, T. (2019). Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing, 338*, 34–45. https://doi.org/10.1016/j.neucom.2019.01.103
16. Oza, P., Sharma, P., Patel, S., Adedoyin, F., & Bruno, A. (2022). Image augmentation techniques for mammogram analysis. *Journal of Imaging, 8*(5), 141. https://doi.org/10.3390/jimaging8050141

17. Chen, J., Yang, N., Pan, Y., Liu, H., & Zhang, Z. (2023). Synchronous medical image augmentation framework for deep learning-based image segmentation. *Computerized Medical Imaging and Graphics, 104*, 102161. https://doi.org/10.1016/j.compmedimag.2022.102161
18. Alomar, K., Aysel, H. I., & Cai, X. (2023). Data augmentation in classification and segmentation: A survey and new strategies. *Journal of Imaging, 9*(2), 46. https://doi.org/10.3390/jimaging9020046

Generative Adversarial Learning for Medical Thermal Imaging Analysis

Prasant K. Mahapatra, Neelesh Kumar, Manjeet Singh, Hemlata Saini, and Satyam Gupta

1 Introduction

Currently, most medical practitioners diagnose disorders using computer-aided imagery. Generally speaking, low-resolution photos made it difficult to diagnose several of the disorders. In order to create artificial images as well as their segmented images, the deep convolutional network will be used. These synthesized photos have a high resolution.

Generative adversarial networks (GANs) have emerged, offering new technologies and a framework for the use of medical pictures. GANs are quickly becoming a cutting-edge foundation as a result of achieving increased performances in a number of medical applications. The technical characteristics of common GAN approaches utilized in the medical imaging domain are extensively elucidated. Unsupervised learning is accomplished using sophisticated neural networks called generative adversarial networks (GANs).

2 What is a GAN (Generative Adversarial Network)?

Generative adversarial networks (GANs), a method for deep learning, allow computers to synthesize new, artificial data from collections of pre-existing data. In particular, a GAN can produce high-quality data with little to no labeling through competition between the generator and discriminator networks [1, 2].

P. K. Mahapatra (✉) · N. Kumar · M. Singh · H. Saini · S. Gupta
Biomedical Applications (BMA) Group, CSIR-CSIO, Chandigarh, UT, India
e-mail: prasant22@csio.res.in; neel5278@csio.res.in; manjeet@csio.res.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_9

There are two competing neural network models in GAN. Using the noise vector (usually a low-dimensional random vector sampled from a normal or uniform distribution typically between 50 and 512 dimensions, and is randomly generated for each sample during training) as an input, one creates samples (and so named generator). The purpose of the noise vector is to introduce randomness into the generator network and to allow it to produce a diverse set of 2 outputs. By providing different random vectors as input to the generator network, we can generate a wide range of new data. In order to ensure that the generated outputs are diverse and not just copies of the training data, the noise vector is an important factor in the success of GANs.

The second model, referred to as the discriminator, is given samples from the generator and training data [3]. The generator has been trained to make images that closely resemble actual data, while the discriminator has been trained to completely distinguish between produced data and true data. The adversarial network's generator and discriminator compete against one another until symmetry is established, at which point the network is trained.

2.1 Overview of GAN Structure

GANs compete two neural networks against each other to establish the probability distribution of a dataset. GAN has two neural networks in it:

- Generator, G.
- Discriminator, D.

A generative network seeks to create artificial images that appear realistic. It accepts a random vector as input (let us say a 100-dimensional array of numbers from a Gaussian distribution) and produces a highly realistic image that appears to be a part of our training set.

On the other hand, the discriminator network accurately determines if an image is fake (i.e., created by the generator) or real (i.e., direct from the source of the input). These processes are repeated many times, so that the generator and the discriminator get better and better at their respective roles with each iteration. Fig. 1 will help you understand how it works.

2.2 Mathematical Equation

The discriminator examines generated images and real images (i.e., training samples) separately. It determines if the discriminator's input image is fake or real. The probability that the input x is real is represented by the output D(x). The discriminator is trained in the same manner as a deep network classifier. We want D(x) = 1 if the input is true, that is, image is real. It should be zero if it is a generated image.

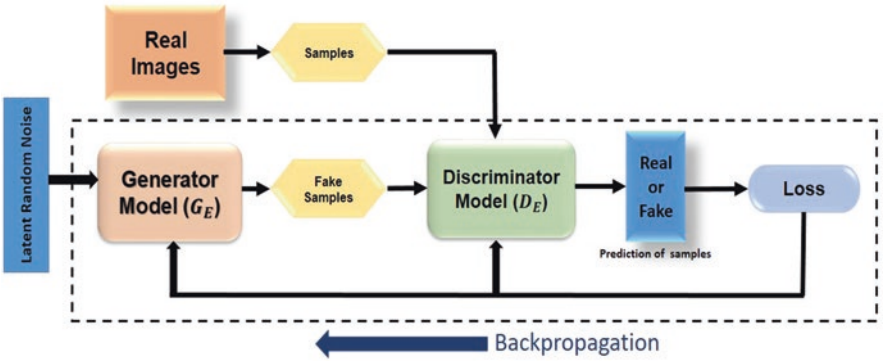


Fig. 1 An illustration of a generative adversarial network (GAN)

The discriminator finds qualities that contribute to realistic images through this method. On the other hand, we want the generator to produce images that are identical to the true image, with D(x) = 1. It backpropagates the desired value all the way back to the generator in order to train the generator to generate images that are more similar to what the discriminator recognizes as real.

The generator becomes stronger at producing realistic images that the discriminator cannot tell apart from actual ones as the training goes on. The discriminator also grows stronger at picking up even the smallest variations between the two sorts of images. The generator eventually creates visuals that are similar to real images as the two models converge.

The following formula can be used to mathematically explain it [4]:

$$\min_{G} \max_{D} V(D,G) \tag{1}$$

$$V(D,G) = E_{x \sim p_{data}(x)} \left[\log D(x) \right] + E_{z \sim p_z(z)} \left[\log \left(1 - D(G(z)) \right) \right]$$

where,

x = real data,
z = noise vector,
G(z;θ_g) = The generator network operates by performing a mapping function from the noise vector to a synthetic data point, with the parameters of the generator network denoted as θ_g.
D(x;θ_d) = The discriminator network is a function that receives a data point as its input and generates a scalar output that denotes whether the input is authentic or artificial. The parameters of the discriminator network are represented by θ_d.
p_data(x) = The probability distribution of the actual data.
p_z(z) = The probability distribution function of the noise vector z.

The principal objective of GANs is to enhance the discriminative capacity of the discriminator in discerning genuine and synthesized images. The aforementioned

process is accomplished via a minimization-maximization methodology, wherein the generator endeavors to minimize its objective, while the discriminator strives to maximize it. The primary aim of the objective function is to enhance the likelihood of detecting artificially generated images as counterfeit and genuine images as authentic, thereby optimizing the likelihood of observed data. The cross-entropy function is a widely adopted method for computing the loss in deep learning, which involves the calculation of p multiplied by the natural logarithm of q. In the context of real images, the appropriate label to assign is p, which has a value of 1. In the case of generated images, the label is inverted, specifically by subtracting it from one. GANs are commonly characterized as a minimax game in which the objective of the generator is to minimize the value of V, while the discriminator aims to maximize it.

2.3 Major Applications of GAN

Wherever new, plausible data is required, GANs can be used in a wide range of applications. GANs are specifically used to produce new images and videos.

Image Generation: GANs are commonly used for generating realistic images. For example, they can be used to generate realistic-looking faces, landscapes, or even artwork.

Style Transfer: GANs have the potential to facilitate the transfer of style from one image to another, thereby enabling the creation of an entirely new image that incorporates the content of one image and the style of another.

Data Augmentation: GANs can be used for generating new data from existing data, which can be useful for training machine learning models with limited datasets.

Disease Diagnosis and Prediction: GANs can be used for identifying patterns in medical data and predicting the likelihood of a patient developing a certain disease.

Medical Image Analysis: GANs can be used for generating synthetic medical images, such as CT or MRI scans, which can be used for training machine learning models. GANs can also be used for image segmentation, enhancing the quality of medical images, and reducing image noise.

Medical Data Augmentation: GANs have the potential to generate synthetic medical data, thereby serving as a means of augmenting limited datasets and enhancing the precision of machine learning models.

Overall, GANs have the potential to revolutionize the field of medicine, by improving the accuracy of disease diagnosis, speeding up drug discovery, and enabling personalized treatment plans.

3 Self-supervised Generative Adversarial Learning

We will first define the term "Self-Supervised Learning" and then discuss how it enhances GANs. Self-supervised is the most similar to unsupervised learning when compared to the prominent families of supervised and unsupervised learning. An effective method for learning representations from unlabeled data is self-supervised (SS) learning [5]. Self-supervised learning algorithm learns from data itself, with no data labeled examples. The algorithm must identify patterns within the dataset to facilitate the process of acquiring knowledge from it [4].

With the help of pseudo-labels [5], self-supervised approaches enable the classifier to learn better feature representation [6]. These methods specifically suggest learning the model to recognize a geometric transformation that has been done to the input image in order to learn an image feature.

There exist several approaches to the implementation of self-supervised learning. One approach to comprehending the attributes of the data is to employ a neural network. Subsequently, the neural network can be employed to forecast the designations of novel data. The identification of data structure can also be accomplished through the utilization of a Convolutional Neural Network (CNN). A CNN can be utilized to forecast the outcomes of novel data.

There are some situations where self-supervised learning is superior to supervised learning. For example, a CNN trained just through self-supervised learning can classify images more accurately than a CNN taught only through supervised learning. This is due to the fact that a CNN that is learned only through supervised learning is limited by the training set that is made accessible to it. A CNN that has been trained only through self-supervised learning can understand the data's structure from scratch, improving its ability to generalize to new data [6, 7].

4 Conditional and Unconditional GANs

The issues with training GANs will now be linked to self-supervised learning. GANs are a type of unsupervised generative modelling in which you may just input data and let the model generate false data from it. Modern GANs, on the other hand, use a method called conditional-GANs [8], which convert the generative modelling challenge into a supervised learning task that needs labeled data. For easier generative modelling, conditional-GANs incorporate class labels within the generator and discriminator [9].

The term "unconditional GANs" eliminates the necessity for class labels in generative modelling. This chapter will demonstrate how self-supervised learning tasks can do away with labeled data when using GANs.

5 Thermal Imaging Systems

The surface skin temperature [10] can be measured using thermal imaging devices. These systems might contain a temperature reference source in addition to an infrared thermal camera [11, 12].

The surface skin temperature of a subject may typically be measured reliably by thermal imaging devices without being in immediate contact to the subject under evaluation [13]. Thermal imaging systems [14] have advantages over other techniques of measuring temperature since they require a closer proximity or touch (Fig. 2).

5.1 Why are Thermal Imaging Devices Beneficial?

There are various advantages of using thermal imaging systems/cameras, which are listed below:

1. *100% non-invasive:* The proximity of the evaluator to the subject under scrutiny is not a requisite for the operation of thermal imaging devices.
2. *Speed and accuracy:* In contrast to traditional forehead or oral thermometers that require close proximity or physical contact with the subject under evaluation, thermal imaging devices have the potential to provide more rapid and precise monitoring of surface skin temperature.
3. *Flexible and cost-effective diagnostic approach:* Thermal camera should be readily available on the market at reasonable price and the same equipment is used to record both thermal and geometric data.

Fig. 2 Shows how to set up thermal imaging properly to analyze persons individually

6 Need of Data Augmentation in GANs

Generating annotated medical imaging data is a challenging and costly task. The creation of deep learning models that can be generalized necessitates the acquisition of substantial amounts of data. Standard data augmentation is a commonly employed technique aimed at enhancing the generalizability of machine learning models. Generative adversarial networks offer a novel method for data augmentation [6, 7].

Insufficient data during the training of GANs often leads to the issue of discriminator overfitting [15] which in turn causes the training process to diverge. Our proposed approach involves utilizing an adaptive discriminator augmentation technique that effectively enhances the stability of training in scenarios where data availability is limited [16, 17]. This approach is applicable for both initial training and does not necessitate modifications to either loss functions or network architectures. The utilization of unlabeled data holds significant value in the improvement of deep learning efficacy. GANs are a potent category of neural networks capable of generating lifelike novel images based on unannotated source images [15, 18]. GANs have been employed in the past to augment data, including the creation of supplementary training images for classification purposes and the enhancement of synthetic images [19].

In order to overcome overfitting and underfitting [2], data augmentation with GANs was demonstrated to boost model accuracy and decrease model loss, hence enhancing the generalizability of the model [20] (Fig. 3).

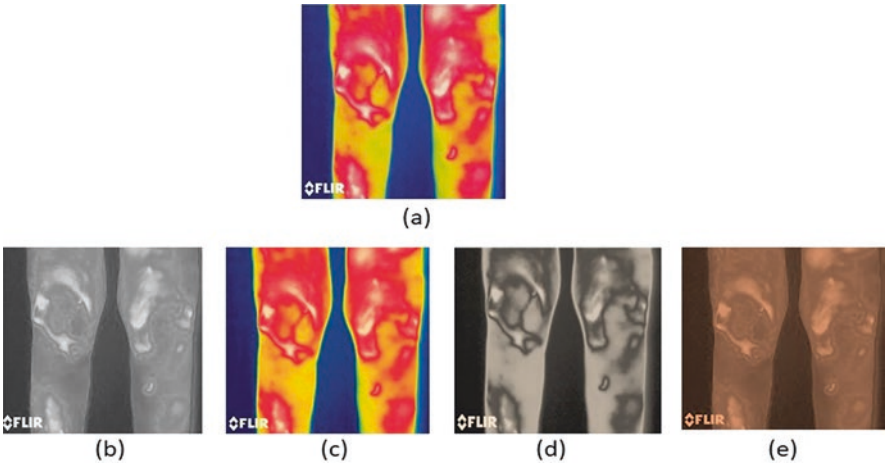


Fig. 3 Thermal image of knee osteoarthritis patient (**a**) and its augmented GAN-generated images (**b**–**e**)

7 Improved Medical Image Generation via Self-supervised Learning

In the domain of deep learning, it is customary to utilize extensive labeled datasets to effectively train a deep neural network. Various self-supervised learning techniques have been suggested as a means to acquire universal visual characteristics in an automated manner, thereby circumventing the laborious and time-intensive process of manually annotating vast quantities of data. Self-supervised generative adversarial neural networks, also known as unconditional GANs, are utilized for the purpose of generating synthetic thermal images.

The widespread use of deep CNNs in computer vision applications can be attributed to their remarkable ability to extract features from visual data. These applications include but are not limited to image classification, semantic and instance segmentation, object recognition, and image captioning. The efficacy of deep learning models is notably impacted by the quantity of data utilized during the training process, as they have the ability to expand and enhance in intricacy with the incorporation of supplementary training data.

8 Methods

Despite the prevalence of comprehensive color image databases for diverse objects in the public sphere, there exists a dearth of comparable databases for thermal images, with either a lack of availability or restricted representation of object categories. The synthesis of thermal images is of great significance due to the arduous nature and high expenses associated with obtaining authentic data. The process of gathering and annotating extensive datasets comprising millions of images is arduous, costly, and time-intensive [21].

GANs have demonstrated remarkable efficacy in generating diverse images through the use of pre-existing images and stochastic noise, a widely acknowledged fact. Currently, unconditional GANs have the ability to generate images that exhibit a high degree of realism, diversity, and quality.

8.1 Training Dataset

The selection of an appropriate unlabeled dataset is an essential part of transfer learning via self-supervised pre-training.

Our training dataset [22] is based on the knee areas [23] of the human body which are captured with a FLIR thermal camera. While diagnosing arthritis, thermography is frequently used to examine deep-bodily joints that are challenging to evaluate with a standard X-ray [19, 24]. The size of all thermal images is 312 KB with dimensions of 320 × 240 pixels [25] (Fig. 4).

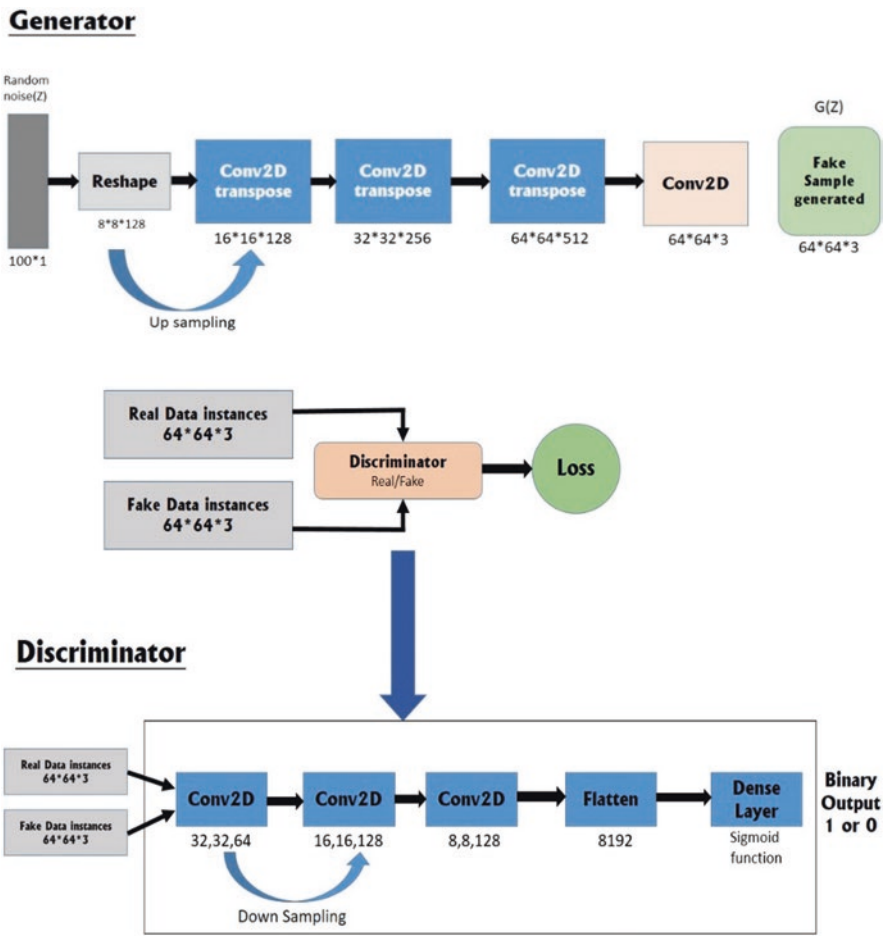


Fig. 4 Generator and discriminator models used in this technique

8.2 Results and Conclusion

The application of thermal imaging technology [26, 27] is employed for the purpose of diagnosing infectious skin conditions and investigating a wide range of disorders, wherein alterations in body temperature may indicate the presence of inflammation in injured tissues or clinical abnormalities that result in changes in blood circulation [23].

The results of the current study indicate that thermal imaging has the potential to serve as a dependable diagnostic modality for detecting measurable patterns in skin temperatures [28]. It has been shown that changes in pain intensity associated with arthritic, repetitive strain, muscular, and circulatory issues can be correlated with temperature variance [29–31].

We believe that this non-intrusive method makes it possible to find the earliest clinical features, with high reliability [32].

8.3 GAN Results

The GAN generated fake images from the given thermal images of Knee dataset and comparison of generator & discriminator loss on a trained GAN architecture are visualised (Figs. 5 and 6).

8.4 Outlook and Conclusions

This chapter has explored various techniques for producing simulated thermal images using the provided knee dataset. Future research in the field of data augmentation will focus on various topics, including the development of a taxonomy of augmentation methods. To improve the quality of GAN samples, researchers may explore novel combinations of meta-learning and data augmentation techniques, investigate the correlation between data augmentation and classifier architecture,

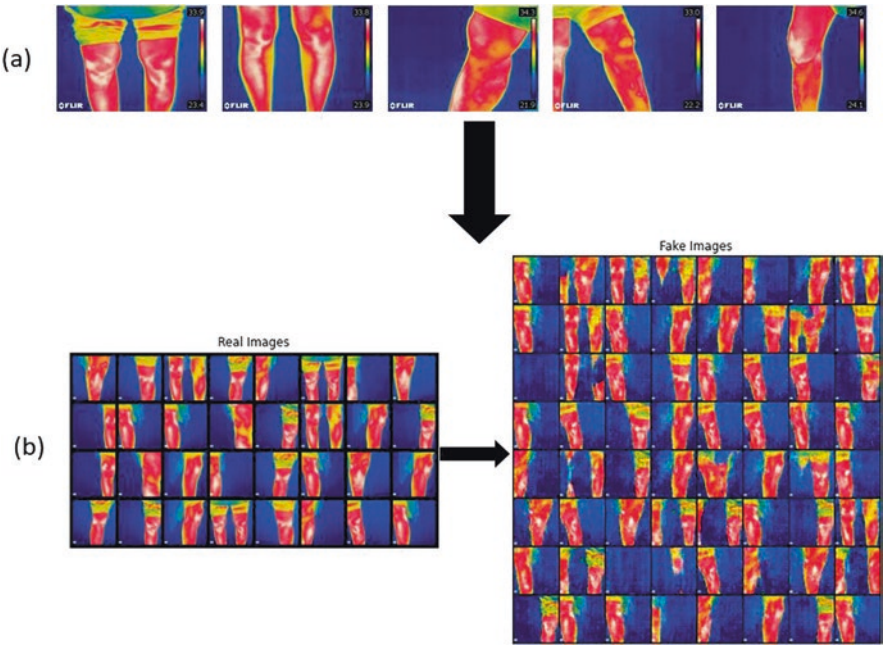


Fig. 5 (**a**) Different thermal images of knee pair and its lateral view (right and left); (**b**) Matrix of input real image and the GAN-generated fake image matrix

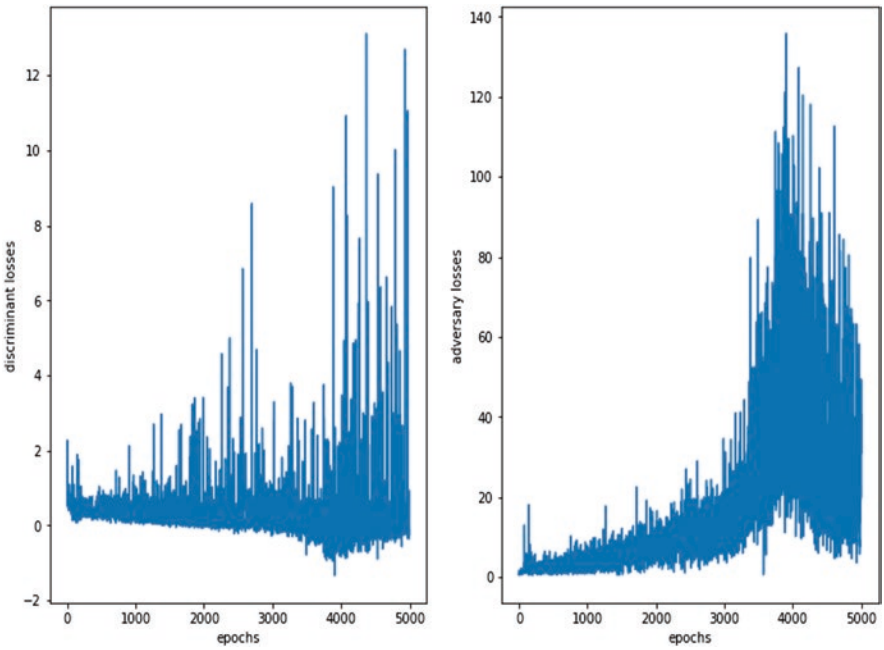


Fig. 6 Comparison of the generator and discriminator loss on a GAN architecture that trained on knee dataset

and apply these concepts to diverse data types. Furthermore, the integration of innovative data augmentation techniques can enhance the variety and magnitude of the training dataset, consequently augmenting the efficacy of the GAN model [33].

In our upcoming study, we intend to investigate performance benchmarks for geometric and color space augmentations on numerous datasets from various image recognition tasks. To show how well these augmentations work in situations when there isn't a lot of data, we are going to impose these dataset's size restrictions. The qualities of the temperature profile that is connected with a thermal image have not yet been investigated while creating synthetic thermal images, which may be a future course of action.

The GAN framework has undergone several modifications in various research articles, utilizing diverse network designs, loss functions, evolutionary techniques, and other methodologies. The study has led to a significant improvement in the quality of samples generated by GANs. An important avenue for further investigation pertains to the augmentation of GANs' sample quality, as well as the assessment of their efficacy across diverse datasets. To advance the exploration of GAN sample combinatorics, we aim to employ supplementary augmentation techniques, including the transfer of diverse styles onto GAN-generated samples.

Future research in generative models with data augmentation should also focus on **StyleGAN2, StyleGAN2-ADA, DiffAugment, and Variational Autoencoder (VAE)**. Trying to produce high-resolution outputs from GAN samples is one of the main challenges. It will be interesting to explore how we might utilize these GAN networks to produce high-resolution images as a result.

Acknowledgments This work is made possible by the core grant (HCP-0026-3.2) of the Medical Mission project, which was sponsored by the Government of India, DSIR, Ministry of Science & Technology, CSIR, India, and CSIR-Central Scientific Instruments Organization, Sector-30, Chandigarh.

References

1. Goodfellow, I., et al. (2020). Generative adversarial networks. *Communications of the ACM, 63*(11), 139–144. https://doi.org/10.1145/3422622
2. Tian, L., Wang, Z., Liu, W., Cheng, Y., Alsaadi, F. E., & Liu, X. (2021). A new GAN-based approach to data augmentation and image segmentation for crack detection in thermal imaging tests. *Cognitive Computation, 13*(5), 1263–1273. https://doi.org/10.1007/s12559-021-09922-w
3. Chen, H. (2021). Challenges and corresponding solutions of Generative Adversarial Networks (GANs): A survey study. *Journal of Physics: Conference Series, 1827*(1). https://doi.org/10.1088/1742-6596/1827/1/012066
4. Treneska, S., Zdravevski, E., Pires, I. M., Lameski, P., & Gievska, S. (2022). GAN-based image colorization for self-supervised visual feature learning. *Sensors, 22*(4). https://doi.org/10.3390/s22041599
5. Taherkhani, F., Dabouei, A., Soleymani, S., Dawson, J., & Nasrabadi, N. M. (2021). Self-supervised Wasserstein pseudo-labeling for semi-supervised image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12267–12277).
6. Araslanov, N., & Roth, S. (2021). Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15384–15394). Available: https://github.com/visinf/da-sac.
7. Song, J., Li, P., Fang, Q., Xia, H., & Guo, R. (2022). Data augmentation by an additional self-supervised CycleGAN-based for shadowed pavement detection. *Sustain., 14*(21). https://doi.org/10.3390/su142114304
8. Elaraby, N., Barakat, S., & Rezk, A. (2022). A conditional GAN-based approach for enhancing transfer learning performance in few-shot HCR tasks. *Scientific Reports, 12*(1). https://doi.org/10.1038/s41598-022-20654-1
9. Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. Class project for Stanford CS231N: Convolutional neural networks for visual recognition. *Winter Semester, 2014*(5), 2.
10. Ludwig, N., Formenti, D., Gargano, M., & Alberti, G. (2014). Skin temperature evaluation by infrared thermography: Comparison of image analysis methods. *Infrared Physics & Technology, 62*, 1–6. https://doi.org/10.1016/j.infrared.2013.09.011
11. Snekhalatha, U., Rajalakshmi, T., & Gobikrishnan, M. (2018). Automated segmentation of knee thermal imaging and X-ray in evaluation of rheumatoid arthritis. *International Journal of Engineering & Technology, 7*, 326–330.
12. Gizińska, M., Rutkowski, R., Szymczak-Bartz, L., Romanowski, W., & Straburzyńska-Lupa, A. (2021). Thermal imaging for detecting temperature changes within the rheumatoid foot. *Journal of Thermal Analysis and Calorimetry, 145*(1), 77–85. https://doi.org/10.1007/s10973-020-09665-0

13. Fernández-Cuevas, I., et al. (2015). Classification of factors influencing the use of infrared thermography in humans: A review. *Infrared Physics and Technology, 71*, 28–55. https://doi.org/10.1016/j.infrared.2015.02.007
14. Mishra, P., & Pathak, K. (2019). *A research paper on thermal imaging system*. [Online]. Available: www.jetir.org.
15. Zhao, S., Liu, Z., Lin, J., Zhu, J. Y., & Han, S. (2020). Differentiable augmentation for data-efficient GAN training. *Advances in Neural Information Processing Systems, 33*, 7559–7570.
16. Zhang, X., Wang, Z., Liu, D., & Ling, Q. (2018). Dada: Deep adversarial data augmentation for extremely low data regime classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2807–2811). IEEE.
17. Maack, L., Holstein, L., & Schlaefer, A. (2022). GANs for generation of synthetic ultrasound images from small datasets. *Current Directions in Biomedical Engineering, 8*(1), 17–20. https://doi.org/10.1515/cdbme-2022-0005
18. Patel, M., Wang, X., & Mao, S. (2020). Data augmentation with conditional GAN for automatic modulation classification. In *WiseML 2020 – Proceedings of the 2nd ACM workshop on wireless security and machine learning* (pp. 31–36), doi: https://doi.org/10.1145/3395352.3402622.
19. Mizginov, V. A., Kniaz, V. V., & Fomin, N. A. (2021). A method for synthesizing thermal images using GAN multi-layered approach. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 44*, 155–162.
20. Shin, H. C., Tenenholtz, N. A., Rogers, J. K., Schwarz, C. G., Senjem, M. L., Gunter, J. L., … & Michalski, M. (2018). Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *Simulation and synthesis in medical imaging: Third international workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018*, Granada, Spain (pp. 1–11). Springer International Publishing.
21. Selfe, J., Hardaker, N., Thewlis, D., & Karki, A. (2006). An accurate and reliable method of thermal data analysis in thermal imaging of the anterior knee for use in cryotherapy research. *Archives of Physical Medicine and Rehabilitation, 87*(12), 1630–1635. https://doi.org/10.1016/j.apmr.2006.08.346
22. Cueva, J. H., Castillo, D., Espinós-Morató, H., Durán, D., Díaz, P., & Lakshminarayanan, V. (2022). Detection and classification of knee osteoarthritis. *Diagnostics, 12*(10). https://doi.org/10.3390/diagnostics12102362
23. Bardhan, S., Nath, S., Debnath, T., Bhattacharjee, D., & Bhowmik, M. K. (2022). Designing of an inflammatory knee joint thermogram dataset for arthritis classification using deep convolution neural network. *Quantitative InfraRed Thermography Journal, 19*(3), 145–171. https://doi.org/10.1080/17686733.2020.1855390
24. Lubkowska, A., & Pluta, W. (2022). Infrared thermography as a non-invasive tool in musculoskeletal disease rehabilitation – The control variables in applicability – A systematic review. *Applied Sciences (Switzerland), 12*(9). https://doi.org/10.3390/app12094302
25. Jin, C., Yang, Y., Xue, Z. J., Liu, K. M., & Liu, J. (2013). Automated analysis method for screening knee osteoarthritis using medical infrared thermography. *Journal of Medical and Biological Engineering, 33*(5), 471–477. https://doi.org/10.5405/jmbe.1054
26. Frize, M., Adéa, C., Payeur, P., Gina Di Primio, M. D., Karsh, J., & Ogungbemile, A. (2011). Detection of rheumatoid arthritis using infrared imaging. In *Medical imaging 2011: Image processing* (Vol. 7962, pp. 205–215). SPIE.
27. Umapathy, S., Vasu, S., & Gupta, N. (2018). Computer aided diagnosis based hand thermal image analysis: A potential tool for the evaluation of rheumatoid arthritis. *Journal of Medical and Biological Engineering, 38*(4), 666–677. https://doi.org/10.1007/s40846-017-0338-x
28. Fokam, D., & Lehmann, C. (2019). Clinical assessment of arthritic knee pain by infrared thermography. *Journal of Basic and Clinical Physiology and Pharmacology, 30*(3). https://doi.org/10.1515/jbcpp-2017-0218
29. Snekhalatha, U., Anburajan, M., Sowmiya, V., Venkatraman, B., & Menaka, M. (2015). Automated hand thermal image segmentation and feature extraction in the evaluation of rheumatoid arthritis. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, 229*(4), 319–331. https://doi.org/10.1177/0954411915580809

30. Snekhalatha, U., Rajalakshmi, T., Gopikrishnan, M., & Gupta, N. (2017). Computer-based automated analysis of X-ray and thermal imaging of knee region in evaluation of rheumatoid arthritis. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, 231*(12), 1178–1187. https://doi.org/10.1177/0954411917737329
31. Suma, A. B., Snekhalatha, U., & Rajalakshmi, T. (2016). Automated thermal image segmentation of knee rheumatoid arthritis. In *2016 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0535–0539). IEEE.
32. Denoble, A. E., Hall, N., Pieper, C. F., & Kraus, V. B. (2010). Patellar skin surface temperature by thermography reflects knee osteoarthritis severity. *Clinical Medicine Insights: Arthritis and Musculoskeletal Disorders, 3*, 69–75. https://doi.org/10.4137/CMAMD.S5916
33. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1). https://doi.org/10.1186/s40537-019-0197-0

Improving Performance of a Brain Tumor Detection on MRI Images Using DCGAN-Based Data Augmentation and Vision Transformer (ViT) Approach

Md. Momenul Haque, Subrata Kumer Paul, Rakhi Rani Paul, Nurnama Islam, Mirza A. F. M. Rashidul Hasan, and Md. Ekramul Hamid

1 Introduction

Recent studies and investigations in the fields of machine learning and deep learning have made significant progress [1] and have emerged as one of the most crucial ones for education [2], performance enhancement, health enhancement, and illness diagnostics [3], offering a range of solutions and simplifying our lives. A brain tumor can develop in any region of the brain or it can be metastatic, indicating that it has spread from another part of the body to the brain. Symptoms such as headaches, seizures, nausea, and cognitive difficulties are commonly associated with brain tumors, although they can vary depending on the tumor's size, location, and type. Chemotherapy, radiation therapy, and surgery are all available treatment modalities for brain tumors. The kind of therapy will depend on the nature and location of the tumor.

Around the world, an expected 308,102 individuals were determined to have an essential cerebrum or spinal cord growth in 2020. In 2020, 251,329 people died due to critical, dangerous brain and CNS growth [4]. Brain cancers represent 85–90% of all essential focal sensory system (CNS) cancers. Brain and sensory system diseases are the tenth leading cause of death in the United States [4].

M. M. Haque · N. Islam
Depratment of Computer Science and Engineering, Bangladesh Army University of Engineering & Technology (BAUET), Natore, Bangladesh

S. K. Paul (✉) · R. R. Paul · M. E. Hamid
Department of Computer Science & Engineering, University of Rajshahi, Rajshahi, Bangladesh

M. A. F. M. Rashidul Hasan
Department of Information & Communication Engineering, University of Rajshahi, Rajshahi, Bangladesh

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_10

There are several challenges associated with obtaining brain tumor datasets. One of the main challenges is the limited availability of high-quality data. Brain tumors are relatively rare, so there may be few cases to study. Additionally, imaging data (such as MRI scans images) is often difficult to acquire and expensive. This can make it challenging to collect a large and diverse dataset. Another challenge is the complexity and variability of brain tumors. Brain tumors can exhibit substantial variations in terms of magnitude, placement, and classification, which present difficulties in constructing a dataset that accurately captures the diverse array of tumor types [5]. Additionally, the imaging data used to diagnose brain tumors can vary, depending on the equipment used, the imaging protocol, and the radiologist's expertise. Lastly, the data is often sensitive, and many privacy concerns must be considered when collecting and sharing medical data. This can make it difficult to obtain patient consent and limit the types of data available for research [5].

The research introduces a solution to this issue by presenting a model named Deep Convolutional Generative Adversarial Network (DCGAN) for generating MRI images. DCGAN, a type of generative model, has the capability to generate images from diverse origins, including MRI scans. The GAN comprises two primary parts: a generator network is employed to generate novel images, while a discriminator network aims to differentiate between the generated images and real images. These two networks are trained in an adversarial manner, where the generator strives to produce images that closely resemble real ones, while the discriminator aims to accurately classify them as either real or generated. From that point onward, the created dataset is utilized for the brain tumor identification model for the proposed work. Besides, the review proposes a transformer-based model called Vision Transformer (ViT) for brain tumor identification. The proposed model exhibits its exceptional performance in contrast to convolutional neural network (CNN)-based models and alternative transfer learning techniques like VGG16, Inception V3, and ResNet50. This superiority is attributed to the enhanced efficacy of the ViT model resulting from the utilization of extensive datasets, surpassing the performance of transfer learning methodologies.

Following is the breakdown of the remaining portions of the paper: While Sect. 2 provides a summary of the relevant research, Sect. 3 provides a full explanation of the suggested system. Section 4 covers the experimental findings and offers a thorough analysis, and Sect. 5 brings the paper to a conclusion with important observations.

2 Related Works

This section represents a literature review of recent works on brain tumor detection using DCGAN and the ViT. We also present some transfer learning-based detection approaches to validate our methods.

Christine Dewi et al. [6] demonstrated that the utilization of DCGAN to generate synthetic data proved effective in enhancing the original dataset, leading to a

notable enhancement in the accuracy of the traffic sign recognition model. They conducted a comparative analysis of various CNN models, such as ResNet50 and DenseNet. The findings indicated that DenseNet outperformed other models in terms of accuracy and computational efficiency, suggesting its potential as a valuable resource for data augmentation in diverse computer vision tasks.

Qiufeng Wu's et al. [7] demonstrated a data augmentation technique based on DCGANs for tomato leaf disease identification. The authors showed the use of DCGANs to generate new images that are similar to the original images but with minor variations. They then combined the generated images with the original ones to form a larger dataset for training deep neural networks. The proposed technique is evaluated on a dataset of tomato leaf images using the GoogLeNet architecture. The results showed that the augmented dataset generated by the DCGANs improved the accuracy of the GoogLeNet model, achieving a top-1 average detection accuracy of 94.33%.

Viola et al. [8] presented a new technique called Fault Face that was introduced to detect ball-bearing failure using DCGAN method. In this approach, synthetic images of ball bearings with different fault types were generated using DCGANs and utilized to train a CNN for classification. The authors assessed the effectiveness of Fault Face on a dataset comprising vibration signals from ball bearings with various faults, including inner race, outer race, and all faults. The results demonstrated that the proposed method surpassed several existing approaches, achieving an impressive overall accuracy of 98.4%.

A. Chughtai et al. [9] explored the application of DCGANs in generating synthetic brain tumor images for medical imaging purposes. They acknowledged that brain tumors contribute significantly to global mortality, and the scarcity of large-scale medical imaging datasets poses a challenge for developing accurate predictive models. In their study, the authors proposed utilizing DCGAN to generate synthetic brain tumor images and conducted a comparative analysis with other Generative Adversarial Networks (GANs). The authors suggested that this approach could potentially serve as a valuable method for augmenting real-world medical image data, leading to improved prediction models.

S. Deepak et al. [10] suggested a technique for categorizing brain cancers using CNN and transfer learning approach. The suggested approach extracted deep features from the MRI scans of brain tumors using the pre-trained models VGG16 and InceptionV3. The Support Vector Machine (SVM) classifier then uses these deep characteristics to categorize data. The study showed that the proposed technique outperforms previous best in class methods in categorizing brain tumors into four distinct categories, with an overall accuracy of 98.69%. The findings suggested that classifying brain cancers using transfer learning with pre-trained CNN models might be successful.

R. Chelghoum et al. [11] proposed a transfer learning approach for brain tumor classification using MRI images and the CNN architectures. The study employed a custom dataset consisting of four types of brain cancers (meningioma, glioma, pituitary, and no tumor) and fine-tuned three pre-trained CNN models (VGG16, InceptionV3, and ResNet50). The authors sourced the datasets from the publicly

accessible The Cancer Genome Atlas (TCGA) dataset. According to the analysis, the refined ResNet50 model exhibited higher accuracy (97.46%), AUC-ROC (99.75%), and F1-score (97.49%) compared to the other models.

The research papers mentioned above all focus on the utilization of DCGAN for brain tumor detection. It is well established that there are various approaches for identifying brain tumors, including those based on widely used CNN architecture. Additionally, recent studies have highlighted the occasional superior performance of the ViT technique compared to traditional CNN models.

2.1 Existing Data Augmentation Algorithms

Some alternative methods are compared to DCGAN for data augmentation techniques. We mention some algorithms and find significant weaknesses compared to the GANs.

To discover a probabilistic encoding for the data, a variational autoencoder (VAE) is a generative model. An encoder network and the decoder network make up its two primary parts. The decoder network maps the dormant space back to the initial information space after the encoder network maps the information to an inert space. Compared to GANs, VAE produces pictures that are smoother and less crisp, and the produced images resemble the training images more frequently [12].

An autoencoder (AE) is a neural network architecture designed to reconstruct input data. It consists of two essential components: an encoder network and a decoder network. The encoder network transforms the input data into a compressed representation, while the decoder network reconstructs the original data from this representation. While an autoencoder (AE) can be utilized for image reconstruction, it lacks the ability to generate novel images. Instead, it generates images that resemble the patterns observed in the training dataset [13].

The Adversarial Autoencoder (AAE) is a generative model that merges the characteristics of GANs and VAEs. It utilizes the strengths of both GANs and VAEs to effectively generate new data samples while also capturing meaningful latent representations of the input data. The system is composed of three primary elements: an encoder, a decoder, and a discriminator network. While the AAE excels at dimensionality reduction and anomaly detection, it is acknowledged that GANs outperform it in generating novel images [14].

Deep Belief Networks (DBNs) are a type of generative model that rely on stacked Restricted Boltzmann Machines (RBMs). These networks consist of multiple layers of RBMs, with each layer learning a more abstract representation of the data. Unsupervised training methods can be applied to DBNs to effectively train higher layers using the knowledge gained from lower layers. By leveraging the learned features from lower levels, DBNs can establish hierarchical representations of the data, enhancing their ability to capture complex patterns and relationships. However, it is worth noting that DBNs might excel in generating high-resolution images but are challenging to train effectively [12].

It is possible to employ DCGANs to create brand-new pictures that are comparable to the concepts already present in a dataset. Using this, one may artificially inflate the size of a dataset. According to the studies on current data augmentation techniques mentioned above, DCGANs are typically thought to be more effective at producing images that are different from the training images. Compared to other algorithms, DCGANs may produce pictures that are sharper and more lifelike.

2.2 Existing Brain Tumor Detection Algorithms

The utilization of CNNs on brain scan databases, such as MRI images, has emerged as a popular and effective approach in applying deep learning for the purpose of brain cancer detection. The CNN can learn to identify patterns in the images indicative of a tumor. Once trained, the model can classify new images as either containing a tumor or not. Additionally, semantic segmentation can be used to localize the tumor, providing the exact coordinates of the tumor within the scan [15].

Transferring learning is an additional well-liked current strategy. Transfer learning is a deep learning technique that uses a pre-built model—typically created on a massive dataset—as the foundation for new tasks. When there is little data available for the new work, this is especially helpful. Applying a pre-trained CNN that has previously been trained on a big dataset of images (like ImageNet) and fine-tuning it on a dataset of brain scans is one method of applying transfer learning for brain tumor identification. Training the previously learned model on the new dataset while maintaining the bulk of the previously trained weights is known as fine-tuning. This allows the model to quickly learn the specific brain scans' specific features that indicate a tumor [16]. Another approach is to use pre-trained models in semantic segmentation and fine-tune them on the dataset of brain scans to detect the tumor. It is worth noting that transfer learning is a powerful technique, but the quality and size of the dataset are crucial. And it should be validated with clinical data and approved by regulatory bodies before it can be used in clinical practice. There are several popular algorithms used for transfer learning in deep learning, including:

VGG: The VGG network is a constructed on CNN architecture and trained on the ImageNet dataset. It is known for its good performance on image classification tasks and is often used as a starting point for fine-tuning new datasets [17].

ResNet: It is well known that the ResNet (Residual Network) design can train incredibly deep neural networks without encountering the issue of vanishing gradients. It is frequently used to optimize new datasets and was trained on the ImageNet dataset [18].

Inception: Inception architecture is a CNN architecture trained on the ImageNet dataset. It is known for its good performance on image classification tasks and is often used as a starting point for fine-tuning new datasets [19].

In this chapter, the authors propose a transformer approach to detect brain tumors. Transfer learning and transformer architectures are related but different concepts in deep learning. Transfer learning is a procedure where a pre-prepared model,

regularly prepared on a huge dataset, is utilized as a beginning stage for another errand. The thought behind the approach is to use the information gained from one undertaking to further develop execution on a connected errand [20]. This is especially helpful when there is limited information accessible for the new undertaking. In contrast, transformer architectures are a class of neural network architectures first introduced in the paper titled "Attention Is All You Need." These architectures rely on the self-attention mechanism, which enables them to effectively process sequential data. While initially designed for natural language processing tasks, transformer architectures have demonstrated remarkable performance in computer vision tasks as well. Notable examples of transformer architectures include BERT, GPT-2, and RoBERTa, which have been pretrained on extensive text corpora and can be fine-tuned for specific tasks to achieve state-of-the-art results. It is important to note that transfer learning, a technique applicable to various neural network architectures, including Transformers, allows leveraging pretrained models for downstream tasks. In our study, we specifically employ the ViT algorithm, which outperforms both CNN and transfer learning-based approaches in terms of efficiency and training observations.

3 Proposed Methodology's

A proposed system for brain tumor detection using ViT likely involves a few steps. At first, we need to collect the dataset of brain scans (such as MRI images) and annotate them to indicate the presence or absence of a tumor. Then the collected data is pre-processed to ensure it is suitable for the ViT model. Subsequently, the approach involves a series of procedures such as image resizing, intensity normalization, and noise reduction. These operations are performed to ensure the data is appropriately prepared for subsequent analysis. The resulting dataset is then utilized to augment the sample size through various data augmentation techniques, thereby enhancing the diversity and quantity of samples available for further processing. During the data augmentation phase, we employ the DCGAN network to expand the sample size. After that, a suitable ViT model is selected, which is a pre-trained model such as ViT, then fine-tuned on the brain scan dataset. The ViT model is then trained on the brain scan dataset. This involves adjusting the model's parameters to optimize its performance on the task of detecting brain tumors.

The trained model then evaluated on a separate dataset of brain scans to measure its performance (presented in Fig. 1). After the completion of training and evaluation, the next step involves deploying the model within a clinical environment. This deployment process entails seamlessly integrating the model into an established imaging system like an MRI scanner. The objective is to enable automatic classification of brain scans, distinguishing between those that exhibit the presence of a tumor and those that do not.

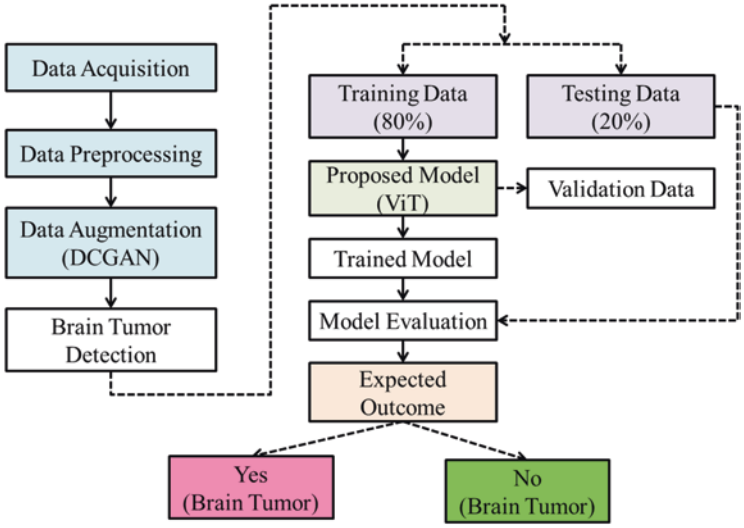


Fig. 1 Proposed system workflow

3.1 Proposed Workflow

The brain tumor detection method we propose comprises several essential steps, which are thoroughly discussed in this section. These steps are instrumental in enhancing the system's performance. The block diagram depicted in Fig. 1 illustrates the proposed system's architecture and the following steps are: Data Acquisition, Data Preprocessing, Data Augmentation, Brain Tumor Detection, and Model Evaluation. The GANS are comprised of three distinct parts: the latent spaces, the generator, and the discriminator.

3.1.1 Data Acquisition

Data acquisition for brain tumor detection using MRI would involve acquiring a large number of MRI scans of the brain [16]. These scans are collected from patients who have been diagnosed with brain tumors, as well as from patients without brain tumors (i.e., healthy controls). To ensure that the dataset is representative of the population, it is important to collect scans from individuals from various age groups, genders, and ethnic backgrounds. Additionally, it is important to collect scans from patients with different types and grades of brain tumors. The collected scans are annotated with information about the presence or absence of a tumor, as well as the location, size, and grade of the tumor if it is present. It is worth noting that data collection and annotation are crucial steps and oftentimes the most time-consuming and complex ones. Therefore, it is important to have a well-designed data collection and annotation plan to ensure the quality and representativeness of the data.

Collecting data from a diverse population is important, but it also increases the complexity of data acquisition, as different patients may have different characteristics and different types of brain tumors. Medical data, such as brain scans or electronic health records, are often stored in different locations and in different formats, making it difficult to access and consolidate the data. Overall, collecting medical data requires a well-designed plan, a dedicated team, resources, and a significant amount of time and effort to ensure that the collected data is representative, high-quality, and compliant with regulatory and ethical standards.

3.1.2 Data Preprocessing

Medical data often presents challenges in terms of noise and inconsistency, making it challenging to ensure data quality. For instance, scans acquired from various machines may exhibit variations in resolution or contrast, posing difficulties in image comparison. In the context of brain tumor detection using ViT, data pre-processing assumes a crucial role. The aim of pre-processing is to prepare the data in a format suitable for ViT while ensuring its high quality. Some common pre-processing methods are used in brain tumor detection. Resizing images to a consistent size can help to ensure that the deep learning model can process the images efficiently [8]. This is especially important for images that have different resolutions or aspect ratios. Normalizing image intensities can help ensure that the deep learning model is not affected by image brightness or contrast. This can be done by converting the images to a standardized scale, such as zero mean and unit variance. To make tumors more visible and minimize picture noise, image enhancing techniques like histogram equalization, contrast stretching, or filtering might be used. The act of finding and eliminating any outliers or inaccuracies from the data is known as data cleaning. This can include removing any images that could be of better quality or have been mislabeled.

3.1.3 Data Augmentation

Data augmentation is a significant methodology for growing the preparation dataset by acquainting irregular changes with the pictures. In the context of brain tumor detection, data augmentation can generate novel images of brain scans featuring tumors with varying locations, sizes, and orientations [21]. One effective method for implementing data augmentation in this domain involves leveraging GANs, specifically DCGANs.

3.1.3.1 DCGAN: Deep Convolutional Generative Adversarial Network

DCGANs belong to the category of GANs and leverage deep convolutional neural networks to produce novel images. Neural generator architecture and neural discriminator architecture make up the two neural architectures that make up DCGANs

[10]. The generator network is prepared to generate brand-new photos that are identical to the authentic photographs in the dataset. Using a succession of transposed convolutional layers, it converts a random noise vector into a picture as input. By changing its loads and predispositions, the generator network figures out how to deliver new images that are equivalent to genuine ones.

The generator network is specifically designed to generate synthetic images that closely resemble the real images present in the dataset. It typically consists of transposed convolutional layers, batch normalization layers, and fully connected layers [22]. A mathematical equation for the output of the generator network can be represented as:

$$\mathrm{G}(z) = f(z,w) \tag{1}$$

In this context, z refers to the random noise vector utilized as an input for the generator network. The set of weights and biases of the network is denoted as w, while $f(z, w)$ represents the function that embodies the neural network. This function encompasses various components such as transposed convolutions, batch normalization, and fully connected layers.

To generate images that closely resemble the authentic images in the dataset, the generator network undergoes a training process. The loss function for the generator network typically comprises both the adversarial loss and the reconstruction loss, which are combined to optimize the network's performance. The loss sustained when the discriminator network can distinguish between the created pictures and the genuine ones is known as the advertisement loss. The unfortunate circumstance known as recreation results in the separation of manufactured pictures from genuine ones (Fig. 2):

$$\mathrm{Loss}(\mathrm{G}) = \mathrm{Loss}(\mathrm{ad}) + \mathrm{Loss}(re) \tag{2}$$

In the context of the given scenario, L (ad) represents the adversarial loss, while L (re) denotes the reconstruction loss. The adversarial loss is the loss that is incurred

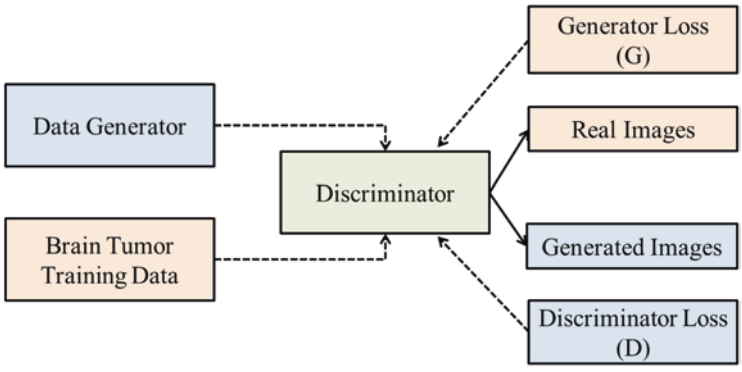


Fig. 2 Proposed architecture of DCGAN algorithm

when the discriminator network can recognize the created images from the real images. The adversarial loss is typically represented as the negative log-likelihood of the discriminator network assigning the label "real" to the generated images. The adversarial loss can be represented as:

$$\text{Loss}(\text{ad}) = -\log\left(\text{D}\left(\text{G}\left(z\right)\right)\right) \tag{3}$$

where D(x) is the output of the discriminator network for the input image x, and G(z) is the output of the generator network for the input noise vector z.

The loss of reconstruction evaluates the difference between the created images and the authentic images. This misfortune is usually estimated utilizing either the Root Mean Square (RMS) or the Structural Similarity Index Measure (SSIM) [18] between the generated and real images. The reconstruction loss can be defined as:

$$\text{Loss}(re) = MSE\left(\text{G}\left(z\right),x\right) \tag{4}$$

In this context, G(z) refers to the output of the generator network when given an input noise vector z, while x denotes the real image. The training parameters for the generator network are outlined in Table 1.

The primary objective in training the discriminator network is to enable it to accurately identify the generated images as authentic representations. It accepts an image as information and determines the likelihood that the image is real or counterfeit. The discriminator network figures out how to recognize the created images from the real images by changing their loads and predispositions [21]. It commonly comprises a progression of convolutional layers, cluster standardization layers, and complete associated layers. The output of the discriminator network is the probability that the input image is real or fake. A mathematical equation for the output of the discriminator network can be represented as:

$$\text{D}\left(x\right) = \text{Sigmoid}\left(f\left(x,w\right)\right) \tag{5}$$

In the context of the neural network, x denotes the input image, w represents the weights and biases of the network, $f(x, w)$ represents the neural network's functional representation, which can encompass convolutional, batch normalization, and fully connected layers. The activation function sigmoid ($f(x, w)$) is responsible for transforming the neural network's output into a probability value ranging between 0 and

Table 1 Model training parameters for a generator network

Model parameters	Value
Total parameter	27, 265, 281
Trainable parameter	27, 265, 281
Non-trainable parameter	0
Activation function	Leaky ReLU, tanh

1. In the given context, x represents the input image, w symbolizes the collection of weights and biases within the network, $f(x, w)$ indicates the neural network's representation, encompassing convolutional, batch normalization, and fully connected layers. In conclusion, sigmoid ($f(x, w)$) means the enactment capability utilized to change over the result of the neural network into a likelihood going somewhere in the range of 0 and 1:

$$\text{Loss}(\text{D}) = -\left(y \log \text{D}\left(x\right)\right) + \left(1 - y\right) * \log\left(1 - \text{D}\left(x\right)\right) \tag{6}$$

where y is the label of the input image (1 for real images, 0 for generated images), and D(x) is the output of the discriminator network for the input image x. The weights and biases of the network are adjusted during the training process to minimize the loss function and improve the discriminator network's abilities [17]. Table 2 represents the model training parameters for a generator network.

The discriminator and generator networks compete for training time. The generator seeks to make pictures that can deceive the discriminator while the discriminator strives to accurately identify the created images. The discriminator becomes better at recognizing the created images as training goes on, while the generator gets better at producing images that look like the genuine photos. The generator network eventually learns to produce new images that are identical to genuine photos.

3.1.4 Brain Tumor Detection

Brain tumor detection using ViT is a transformer-based approach that utilizes the power of ViT models to detect tumors in brain images. Specifically created for image identification tasks, the ViT is a transformer model.

3.1.4.1 Vision Transformer (ViT)

Since it can learn fine-grained features and global relationships between picture patches, the deep learning model architecture known as Vision Transformer (ViT) has become popular and useful for image classification tasks [23]. Specifically created for image identification applications, the ViT is a transformer model. It is based on the transformer architecture, which was created primarily for problems related to natural language processing but was modified for use with image age identification

Table 2 Model training parameters of discriminator network

Model parameters	Value
Total parameter	582,785
Trainable parameter	582,785
Non-trainable parameter	0
Activation function	Leaky ReLU, Sigmoid

applications. The Transformer Encoder, which comprises of a number of multi-head self-attention and feed-forward layers, is a ViT's primary component [9]. ViT learns features more quickly and effectively by using self-attention techniques to record the connections between several picture patches (refer to Fig. 3).

The visual transformer isolates an image into fixed-size patches, accurately installs every one, and incorporates positional inserting to contribute to the transformer encoder. Besides, ViT models beat CNNs by nearly multiple times with regards to computational productivity and exactness [24].

Partition the image into fixed-size patches, standardize the image patches, create lower-level linear representations from these smoothed image patches, integrate positional embeddings, feed the sequence as an input to a cutting-edge transformer encoder, include positional embeddings, feed the sequence as an input to a high-performing transformer encoder, pretrain the ViT model using image descriptors, which is further refined on a comprehensive dataset, and adapt the downstream dataset for image categorization [25]. The transformer encoder is composed of a series of multi-head self-attention layers and feed-forward layers (shown in Fig. 3). Each self-attention layer can be represented mathematically as [26]:

$$\text{Attention}\left(Q,V,K\right) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{7}$$

In this scenario, *Q, K*, and *V* correspond to the query, key, and value matrices, respectively, while d represents the dimension of the key matrix.

$$f\left(x\right) = W_x + b \tag{8}$$

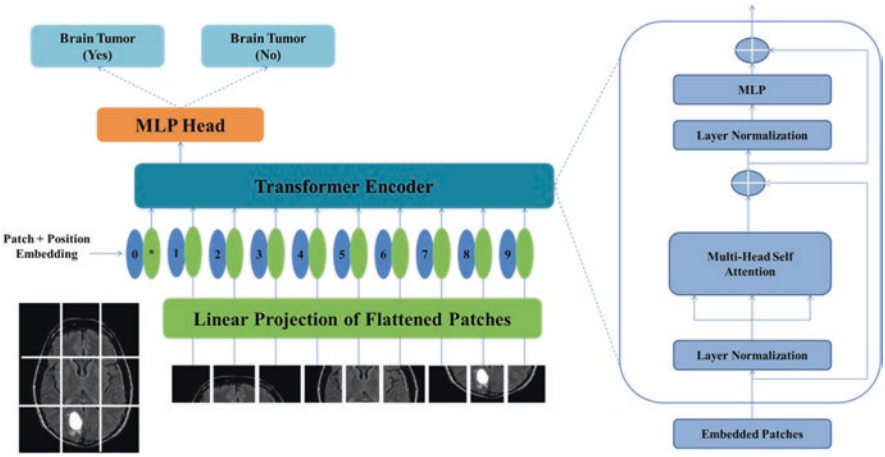


Fig. 3 Vision Transformer (ViT) internal architecture

Here, x denotes the input, W represents the weight matrix, and b indicates the bias vector. The pooling layer is typically implemented as either global average pooling or global maximum pooling. The fully connected layer accepts the pooled feature map as input and generates a probability distribution across the potential classes [27]. This can be represented mathematically as:

$$y = \text{Softmax}\left(W_2\left(\text{Pooling}\, W_1 x + b_1\right)\right) + b_2 \tag{9}$$

where y represents the output of the last layer, which is a probability distribution over the possible classes and x denotes the input image. W_1 and W_2 refer to the weight matrices, while b_1 and b_2 represent the bias vectors. Pooling refers to the pooling function applied in the model.

In ViT, the loss function commonly used is a composite of cross-entropy loss and a regularization term. The cross-entropy loss quantifies the disparity between the predicted class probabilities of the model and the actual class labels. Meanwhile, the regularization term is incorporated to mitigate the risk of overfitting.

In a ViT, an input image is typically divided into several patches, then processed independently [19]. This is done by applying a sliding window operation over the image, where a specific-size window (e.g., 16 × 16 pixels) is moved across the image, extracting a patch at each location. These patches are then fed into the transformer layer for further processing (refers to Fig. 4). Model training parameters of the ViT are demonstrated in Table 3.

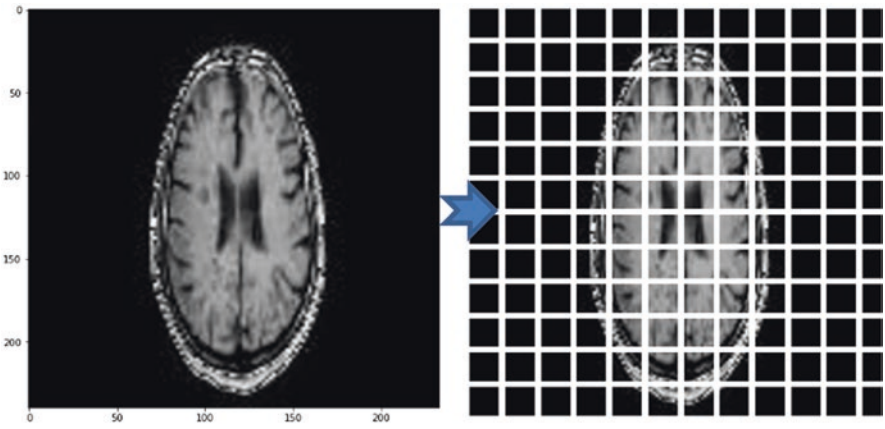


Fig. 4 Input images convert into several patches in a layer

Table 3 Model training parameters of the Vision Transformer (ViT)

Model parameters	Value
Patch size	20
Patches per image	144
Projection dimension	64
Number of head	4

3.2 Model Evolution Metrics

The effectiveness of the ViT model is evaluated using a number of accepted techniques. Accuracy is the proportion of samples that are properly categorized, making it the most fundamental assessment metric. To assess the effectiveness of a classification model, a confusion matrix is a popular table. The output presents the count of predictions produced by the model on a specific dataset, categorized as TP: True Positives, TN: True Negatives, FP: False Positives, and FN: False Negatives.

4 Experimental Results and Discussions

4.1 Dataset Description

Here we use the public brain tumor dataset, actual name is "Brain MRI," which contains 98 normal case images and 155 brain tumor-positive case images from different people (sample images shown in Figs. 5 and 6) [28]. Each image is captured from a different angle and has other characteristics. The dataset is divided into two portions, with 80% allocated for training the model and 20% reserved for evaluating its performance. Furthermore, an additional 10% of the data is designated for validating the model's performance. After that, we apply to resize the images into 128×128 dimensions and normalize the method for data augmentation. In the data augmentation stage, we generate the augmented images to improve the detection model's performance more than before.

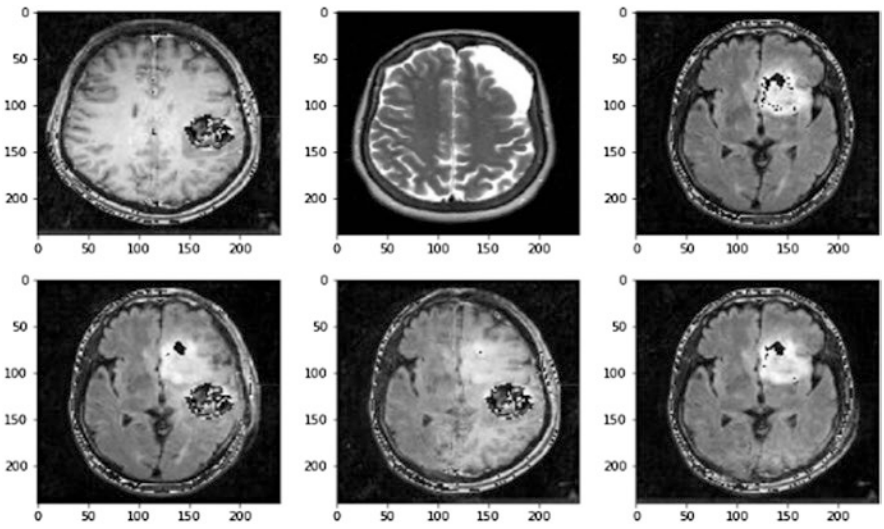


Fig. 5 Brain tumor positive dataset samples

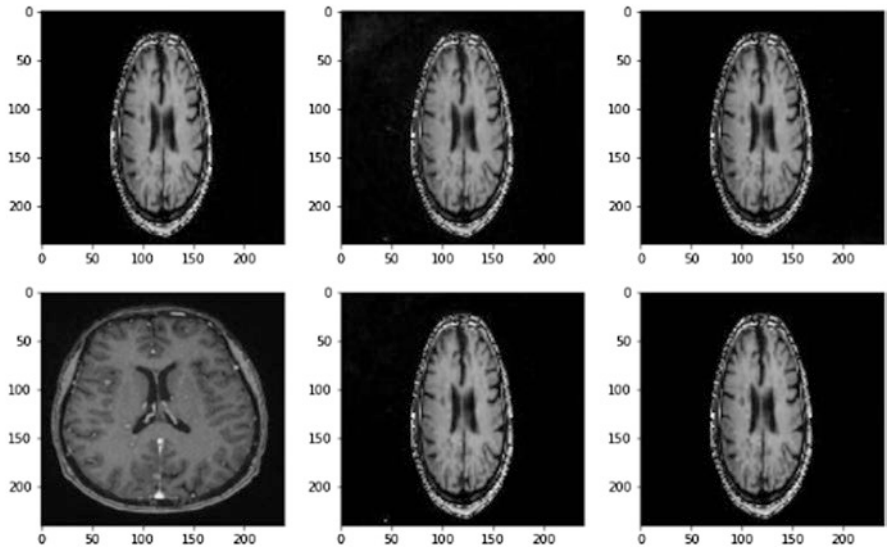


Fig. 6 Brain tumor negative dataset samples

4.2 Data Augmentation Analysis

DCGANs can be used for data augmentation by training the generator architecture to create new images from the preparation information. The generated images can be included in the training set, thereby expanding the dataset and offering supplementary samples for the model to learn from. This augmentation of the dataset aids in enhancing the model's performance and mitigating overfitting by providing diverse and additional training examples [22]. Another way to use DCGAN for data augmentation is to utilize the generator network to create images like the test information and then use them to increase the test set. This can help improve the model's robustness and make it more resilient to variations in the test data. Moreover, there is an option to fine-tune the generator network on a particular dataset, allowing for the generation of new images that exhibit similar characteristics. This approach proves beneficial, especially when working with datasets that have limited data availability. Moreover, it is essential to evaluate the generated images' quality and assess the model's performance on the expanded dataset to validate the efficacy of the data augmentation procedure.

In this data-augmented process, we have generated 1000 brain tumor-positive images and 972 brain tumor-negative images (sample images are illustrated in Figs. 7 and 8). Also, the images differ from the training images, and the loss is pretty high compared to the real images. Now, our dataset becomes 1155 images for brain tumor-positive patients and 1070 images for brain tumor-negative patients.

As we can see from Figs. 9 and 10, the dissemination of created images is almost equivalent to that of real images. In this study, we train a DCGAN on brain MRI

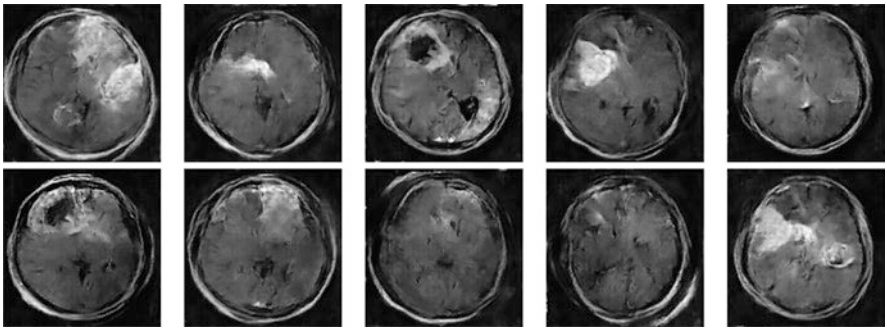


Fig. 7 Generated brain tumor-positive dataset samples through the DCGAN approach

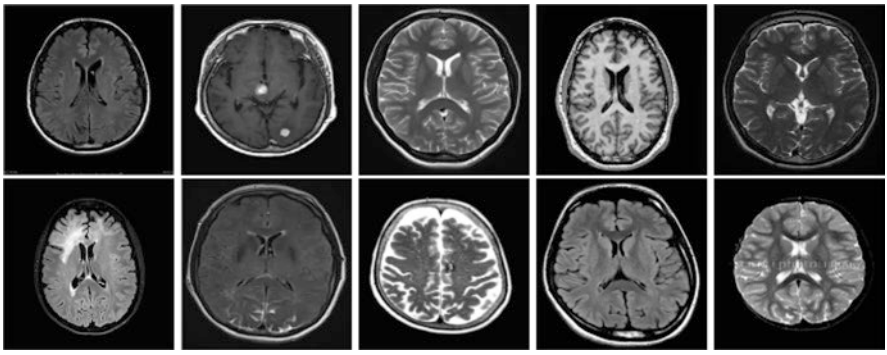


Fig. 8 Generated brain tumor-negative dataset samples through the DCGAN approach

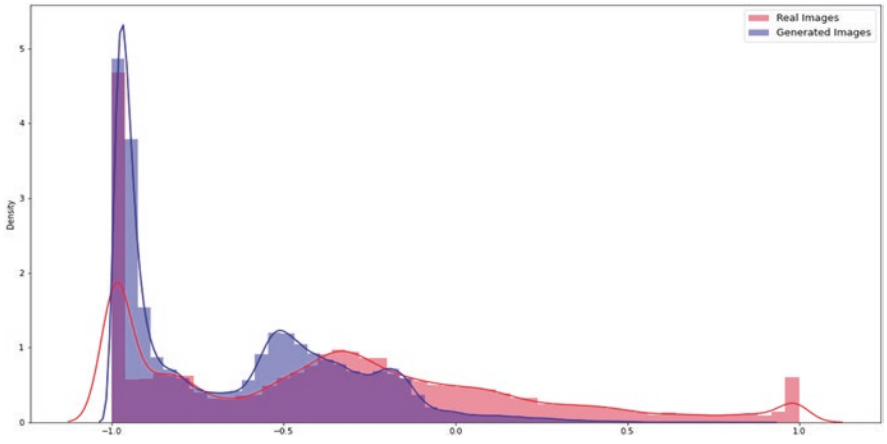


Fig. 9 Density comparison between real images and generated images for brain tumor-positive case

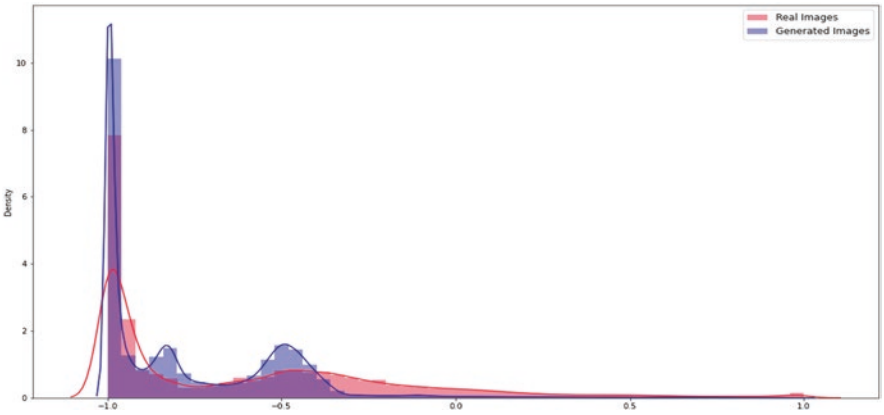


Fig. 10 Density comparison between real images and generated images for brain tumor negative case

images to generate synthetic images of brain tumors. We compare the losses of the DCGAN model for the real world and generated images of both brain tumor positive (in Fig. 9) and negative (in Fig. 10) cases. We compare the density curves of the real and generated images using kernel density estimation (KDE) and plot the results. The density curves of the real and generated images are plotted for both brain tumor positive and negative cases.

For the brain tumor-positive case (in Fig. 9), the density curve of the real images presents a peak at the location of the tumor, indicating a higher density of pixels in that area. However, the density curve of the generated images presents a peak at a different location, indicating that the generated images may not be as accurate as the real images.

For the brain tumor negative case (in Fig. 10), the density curve of the real images presents a relatively uniform pixel density distribution. In contrast, the density curve of the generated images presents a similar distribution with a slight peak at the center of the brain, indicating that the generated images are more accurate and realistic. The difference in the density curves of the real and generated images for brain tumor-positive cases may be due to the complexity of the tumor structure and the limited number of positive cases in the dataset. The generated images may not accurately capture the subtle details of the tumor, resulting in a different peak location in the density curve.

4.3 Classification Performance Analysis

In this section, we discuss the performance metrics for comparing the performance of different machine learning models. Here are some commonly used performance metrics [29] in machine learning:

- *Accuracy Score*: This metric measures the proportion of accurate predictions made by a model out of all the predictions made:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

- *Precision Score*: This metric calculates the proportion of true positives among all the positive predictions made by a model:

$$PS = \frac{TP}{TP + FP} \tag{11}$$

- *Recall Score*: This metric determines the proportion of true positives among all the positive instances present in the dataset:

$$RS = \frac{TP}{FN + TP} \tag{12}$$

- *F1 score*: This metric represents the harmonic mean of precision and recall, providing a comprehensive evaluation of the overall performance of the model:

$$FS = \frac{2 * PS * RS}{PS + RS} \tag{13}$$

- *AUC-ROC Score*: This measures the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate for different classification thresholds. AUC-ROC is a good measure of how well a model can distinguish between positive and negative instances [30].
- *Cohen's Kappa*: It is a statistical measure utilized to evaluate the degree of consensus among two or more raters when rating categorical data. This measure quantifies the level of inter-rater reliability while considering the likelihood of chance agreement. It is expressed on a scale of −1 to 1, where values closer to 1 indicate a significant level of agreement surpassing chance, values close to 0 indicate agreement no better than chance, and values below 0 indicate disagreement surpassing chance expectation [31]:

$$\text{Cohens Kappa} = \frac{P_0 - P_e}{1 - P_e} \tag{14}$$

where

P_o = Relative observed agreement among raters
P_e = The hypothetical probability of chance agreement.

Tables 4 and 5 illustrate the result analysis of different brain tumor detection algorithms and performance parameters of different brain tumor detection algorithms, respectively.

Table 4 shows that the ViT with a DCGAN-based data augmentation algorithm gets the highest accuracy and minimum loss compared to all other models. Moreover, Table 5 also illustrates the superiority of the proposed ViT with a DCGAN-based data augmentation algorithm compared to other well-known models in the literature. Table 6 presents the training parameters for the ViT algorithm we consider in this study. Lastly, Table 7 presents the loss of generating new images from training images using the DCGAN algorithm.

Table 4 Result analysis of different brain tumor detection algorithms

Applied different algorithms	Testing accuracy	Testing loss	Cohen's Kappa	AUC
CNN	96.85%	0.2015	0.937072	0.968602
VGG16	98.12%	0.3022	0.970111	0.985168
ResNet50	95.54%	0.3235	0.958136	0.978957
Inception V3	56.69%	5.2223	0.000970	0.500482
Vision transformer (ViT) (without DCGAN)	86.27%	0.6234	0.714628	0.860484
Vision transformer (ViT) (with DCGAN)	99.33%	0.0610	0.977524	0.988797

Table 5 Performance parameters of different brain tumor detection algorithms

Applied different algorithms	Precision	Recall	F1 score
CNN	0.97	0.97	0.97
VGG16	0.98	0.98	0.98
ResNet50	0.97	0.98	0.97
Inception V3	0.51	0.60	0.55
Vision transformer (ViT) (without DCGAN)	0.86	0.86	0.86
Vision transformer (ViT) (with DCGAN)	0.99	0.99	0.99

Table 6 Training parameters for the proposed Vision Transformer (ViT)

Parameters	Value
Batch size	7
Epochs	120
Learning rate	0.001
Weight decay	0.0001
Image size	240 × 240
Optimizer	Adam
Loss function	Sparse categorical

Table 7 Data augmentation losses of the DCGAN algorithm

Class(s) (Brain tumor)	Loss	Value
Yes	Discriminator loss	0.0049
	Generator loss	5.0134
No	Discriminator loss	0.0159
	Generator loss	4.8392

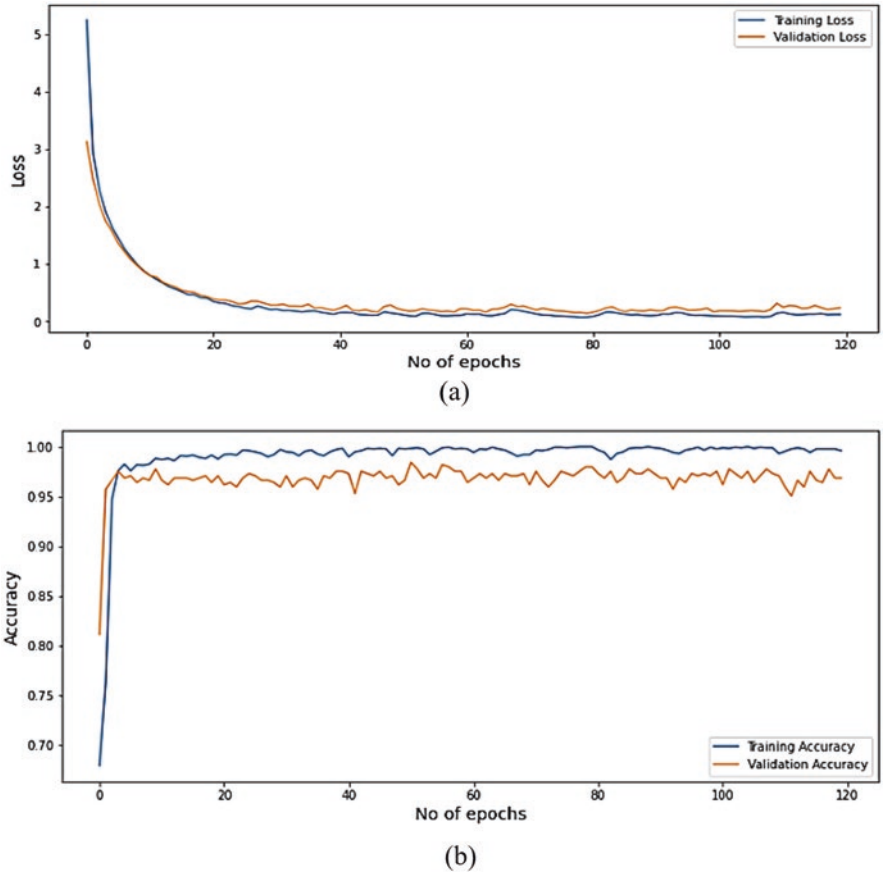


Fig. 11 The figure is presented in (**a**) Validation loss vs training loss, and (**b**) accuracy curve of the CNN algorithm

The different model training and validation loss and accuracy curves are illustrated in Figs. 11, 12, 13, 14, 15, and 16. According to the analysis of Figs. 11–16, the ViT outperformed the other training model, as demonstrated by the model loss and accuracy curve. Specifically, our model achieved a lower loss and higher accuracy than the different models, as evidenced by the visual comparison of the below-mentioned respective curves. This improvement can be attributed to a better

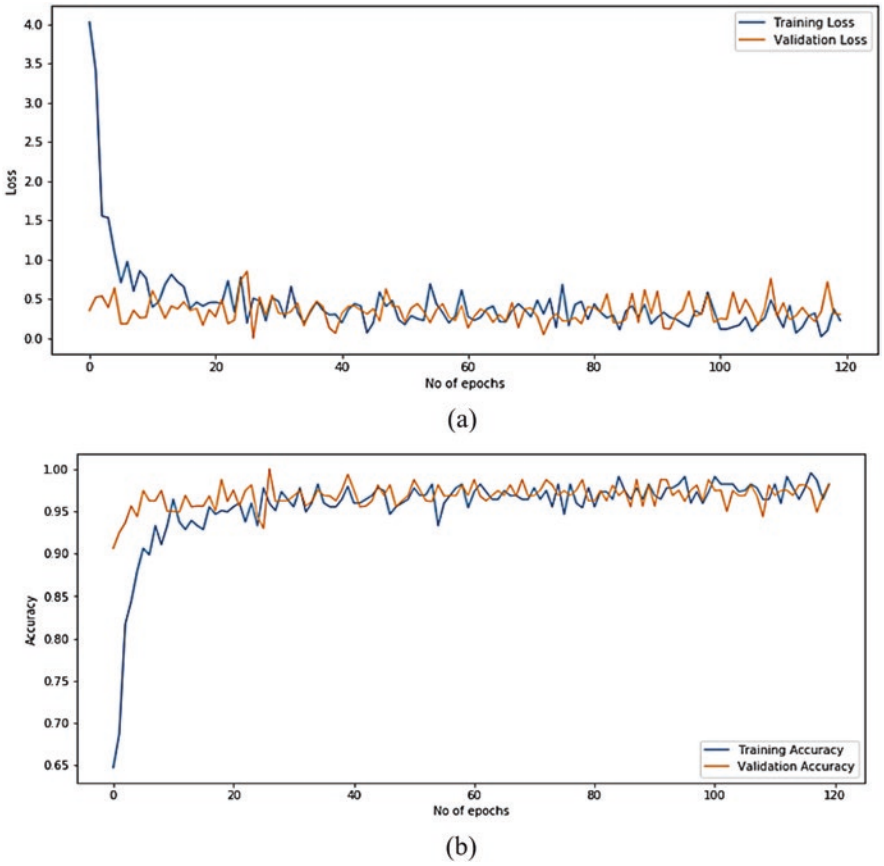


Fig. 12 The figure is presented in (**a**) Validation loss vs training loss, and (**b**) accuracy curve of the VGG16 algorithm

augmentation approach using DCGAN, which allowed our model to learn more effectively from the training data and generalize better to unseen data compared to the CNN, VGG16, ResNet50, InceptionV3, and ViT model without data augmentation using DCGAN approach.

Based on the information provided, it seems that the confusion matrix (Fig. 17) shows that the ViT model with the DCGAN algorithm has the highest accuracy compared to other transfer learning models such as CNN, VGG16, ResNet50, and Inception V3, as well as the ViT model without the DCGAN algorithm. Furthermore, the analysis suggests that the ViT model outperforms both the CNN and transfer learning methods. Increasing the number of samples can improve the performance of the ViT model, and the use of DCGAN with the ViT model can result in better performance in brain tumor detection.

Overall, the ViT model with the DCGAN algorithm has shown promising results for detecting brain tumors and could be used in future applications. However, it is

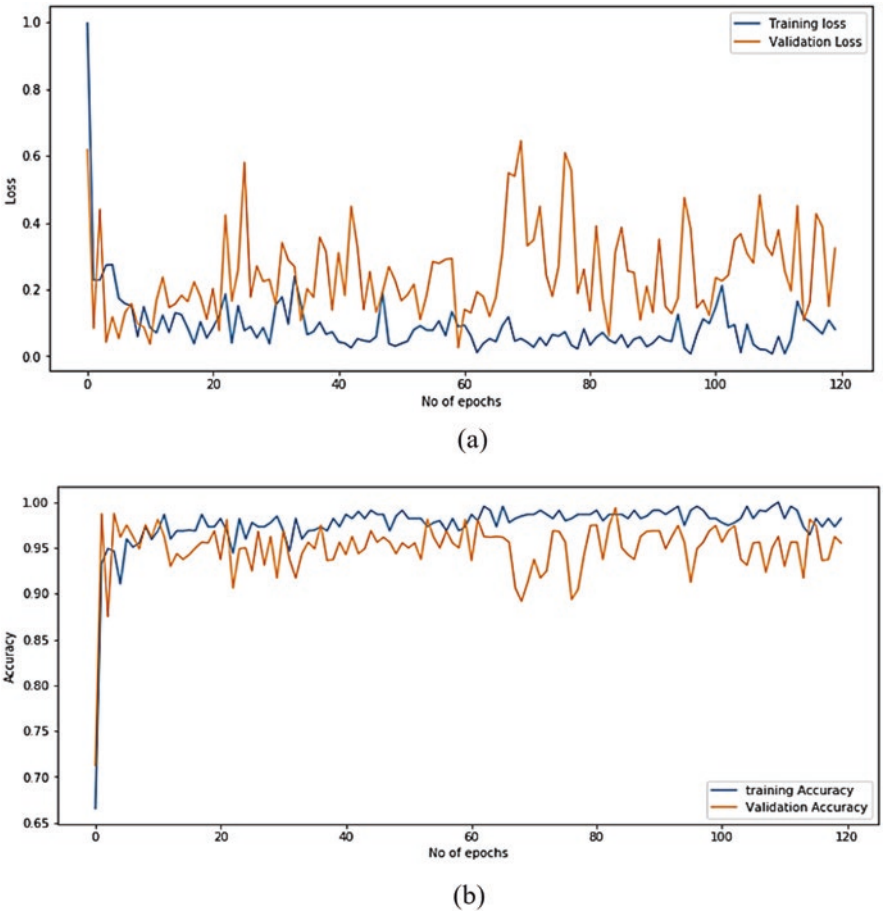


Fig. 13 The figure is presented in (**a**) Validation loss vs training loss, and (**b**) accuracy curve of the VGG16 algorithm

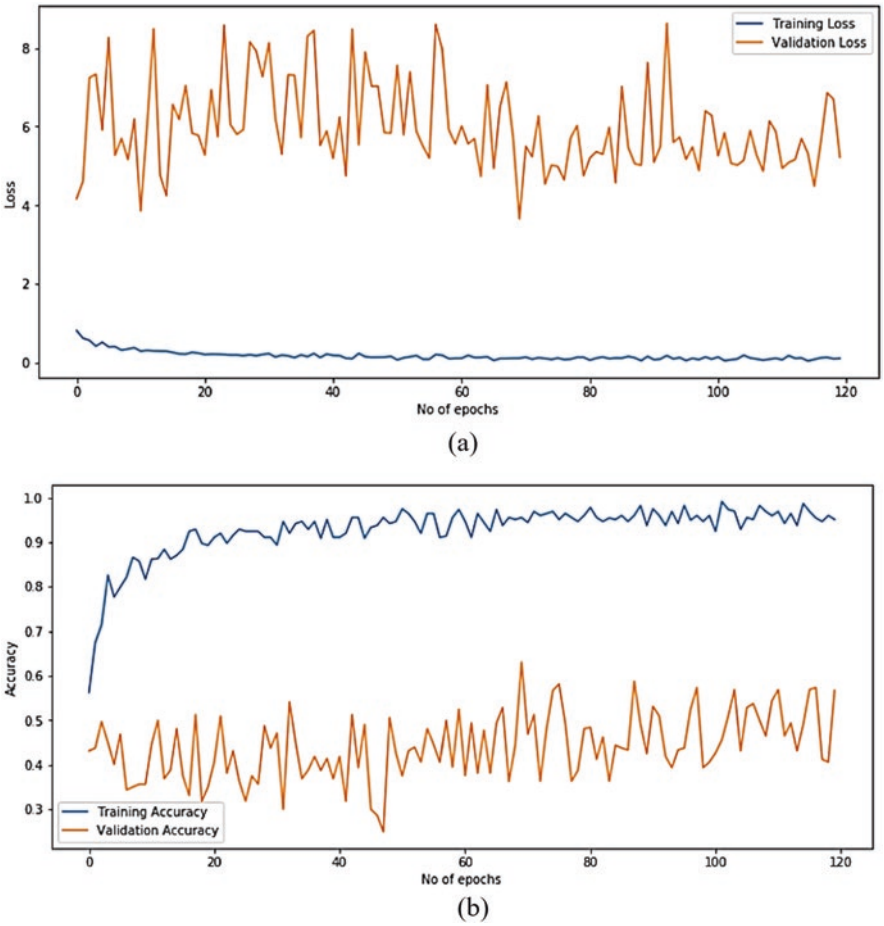


Fig. 14 The figure is presented in (**a**) Validation loss vs training loss, and (**b**) accuracy curve of the Inception V3 algorithm

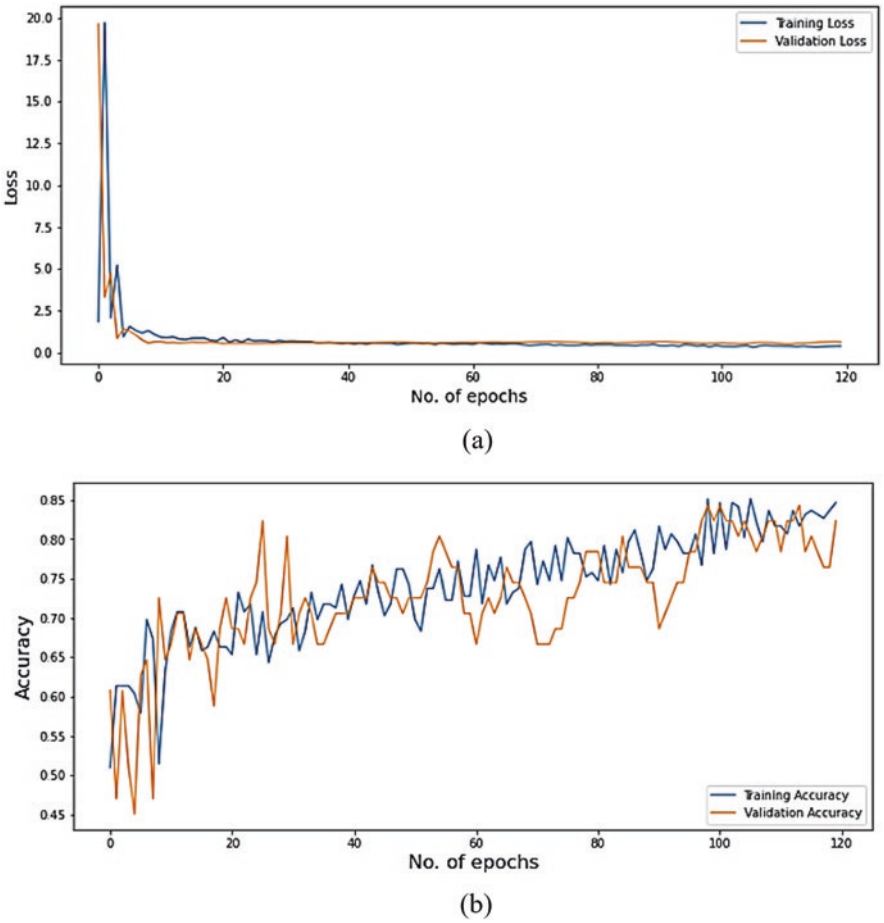


Fig. 15 The figure is presented in (**a**) Training loss vs validation loss, and (**b**) accuracy curve of the Vision Transformer (ViT) before applying the DCGAN algorithm

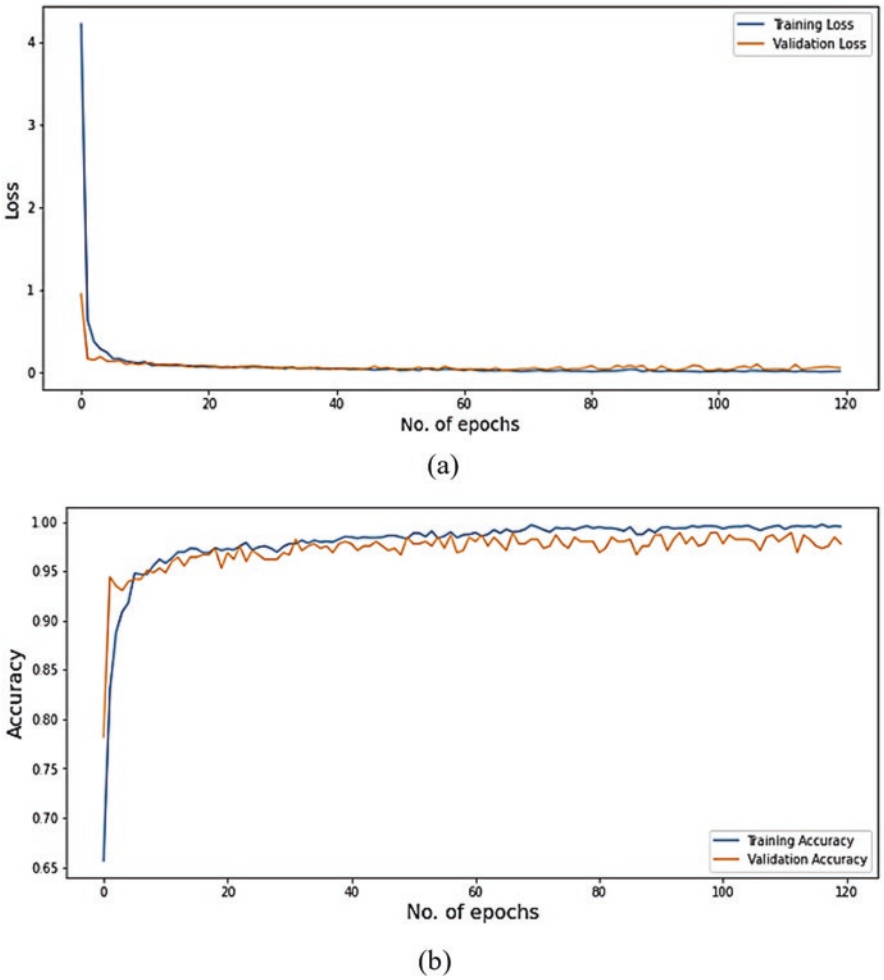


Fig. 16 The figure is presented in (**a**) Training loss vs validation loss, and (**b**) accuracy curve of the Vision Transformer (ViT) after applying the DCGAN algorithm

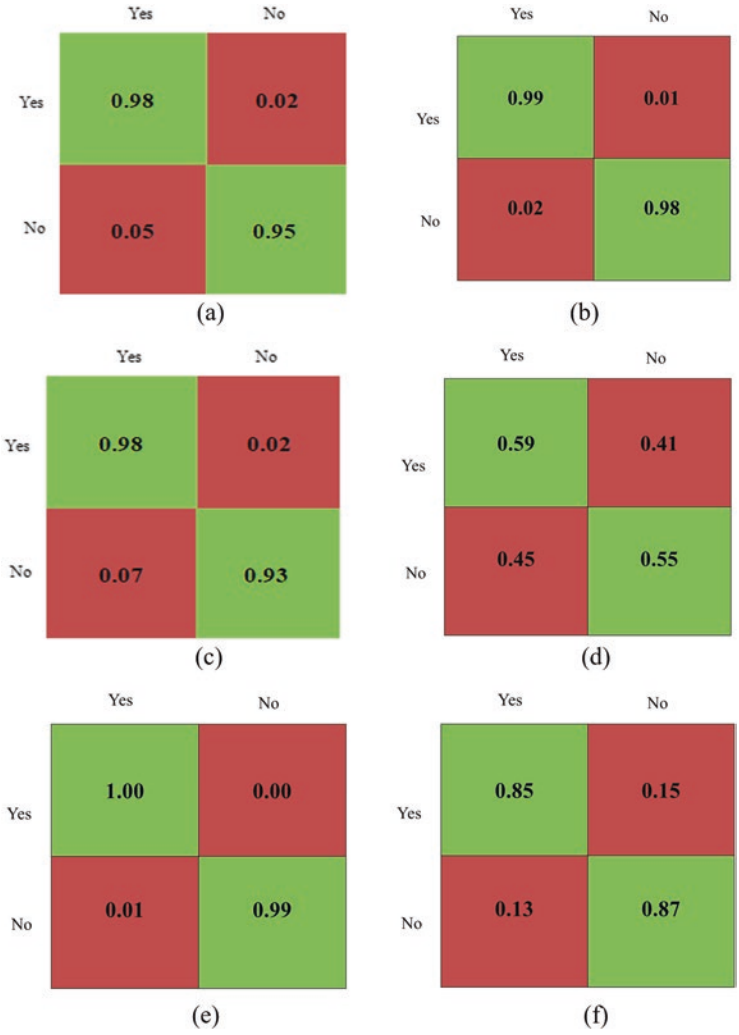


Fig. 17 Normalized confusion matrices were generated for different algorithms including (**a**) CNN algorithm, (**b**) VGG16 algorithm, (**c**) ResNet50 algorithm, (**d**) Inception V3 algorithm, (**e**) Vision Transformer with DCGAN algorithm, and (**f**) Vision Transformer without DCGAN algorithm

essential to keep in mind that the performance of any model depends on various factors, such as the size and quality of the dataset, the choice of hyperparameters, and the specific problem being addressed.

4.4 Comparative Assessment of the Proposed Approach and Established Methods

Our research findings indicate that the proposed model surpassed the performance of the existing model across multiple evaluation metrics, including accuracy, precision, recall, F1-score, and computational efficiency. These results provide compelling evidence that the proposed model offers a more effective solution for brain tumor detection compared to the existing model. Compared to CNNs, ViT has shown promising results in achieving high accuracy on various image classification benchmarks, with some studies showing that it outperforms CNNs on large-scale datasets. However, it is important to note that ViT requires a large amount of training data and computational resources to achieve its full potential.

Our proposed model for brain tumor detection demonstrated better accuracy than an existing model, according to the results presented in Table 8. These findings

Table 8 Accuracy comparative analysis between proposed method and existing method

Author references	Method(s)	Performance score
Deepak and Ameer [32]	Pre-trained GoogLeNet	98%
Majib et al. [33]	VGG-SCNet	Precision: 99.2%, recall: 99.1%, F1-score: 99.2%
Toğaçar et al. [34]	BrainMRNet	96.05%
Zaw et al. [35]	Naïve Bayes	94%
Hossain et al. [36]	CNN	97.87%
Younis et al. [37]	CNN, VGG16	CNN: 96%, VGG 16: 98.5%
Kora et al. [28]	VGG16	98.16%
Fidon et al. [22]	CNN	Accuracy: 84%, precision: 91%, recall: 96%, F1-score: 83%
Liu et al. [25]	G-ResNet	95%
Amin et al. [38]	Inceptionv3	Greater than 94%
Our proposed method	**Proposed: CNN, Proposed: VGG16, Proposed: ResNet50, Proposed: InceptionV3, Proposed: ViT (without DCGAN), Proposed: ViT (with DCGAN)**	**96.85%, 98.12%, 95.54%, 56.69%, 86.27%, 99.33%**

suggest that the proposed model has the potential to be an effective tool for healthcare professionals in detecting brain tumors and that further investigation and validation of the model are warranted to confirm its efficacy.

5 Conclusion

It is possible to use ViT for brain tumor detection, as we have shown it to be effective in image classification tasks. ViT can learn fine-grained features and global relationships between image patches, which can help detect subtle differences in medical images such as MRI scans. Without the DCGAN image-generated approach, ViT performs lower than CNN and other transformer learning approaches. An increasing number of images improves model learning, improves the model's performance, and provides perfect detection results compared to the others.

However, it is worth noting that using ViT for brain tumor detection would require a large dataset of labeled MRI scans to train the model and evaluate its performance. Additionally, it is important to consider the ethical and legal implications of using such models, especially in medical imaging. The model should be validated by experts in the field and meet the standards of regulatory bodies before it can be used in a clinical setting. In summary, ViT can be a promising approach for brain tumor detection. Still, it is important to note that it would require a large dataset, validation, and regulatory approval before being used in clinical practice.

Acknowledgments The Ministry of Posts, Telecommunication, and Information Technology of Bangladesh, The People's Republic of Bangladesh, through the Information and Communication Technology Division, has provided funding for this study in the form of research grants under the ICT-fellowship program.

References

1. Zaman, S. T., Paul, S. K., Paul, R. R., & Hamid, M. E. (2021). Detecting diabetes in human body using different machine learning techniques. In *International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*
2. Paul, R. R., Paul, S. K., & Hamid, M. E. (2022). A 2D convolution neural network based method for human emotion classification from speech signal. In *2022 25th International Conference on Computer and Information Technology (ICCIT)* (pp. 72–77).
3. Paul, S. K., Paul, R. R., Nishimura, M., & Hamid, M. E. (2021). Throat microphone speech enhancement using machine learning technique. In *Learning and analytics in intelligent systems* (pp. 1–11).
4. Huang, J., et al. (2022). The comparative burden of brain and central nervous system cancers from 1990 to 2019 between China and the United States and predicting the future burden. *Frontiers in Public Health, 10*, 3970.
5. Paul, S. K., & Paul, R. R. (2021). Speech command recognition system using deep recurrent neural networks. In *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*.

6. Dewi, C., et al. (2022). Synthetic data generation using DCGAN for improved traffic sign recognition. *Neural Computing and Applications, 34*(24), 21465–21480.
7. Wu, Q., Chen, Y., & Meng, J. (2020). DCGAN-based data augmentation for tomato leaf disease identification. *IEEE Access, 8*, 98716–98728.
8. Viola, J., Chen, Y., & Wang, J. (2021). Fault face: Deep convolutional generative adversarial network (DCGAN) based ball-bearing failure detection method. *Information Sciences, 542*, 195–211.
9. Hassan, M., Malik, R., Arshad, K., & Siddiqui, M. R. U. (2022). Brain tumor image generations using deep convolutional generative adversarial networks: (DCGAN). *Journal of NCBAE, 1*(3), 33.
10. Tummala, S., Kadry, S., Bukhari, S. A. C., & Rauf, H. T. (2022). Classification of brain tumor from magnetic resonance imaging using vision transformers ensemble. *Current Oncology, 29*, 7498–7511.
11. Chelghoum, R. et al., 2020. Transfer learning using convolutional neural network architectures for brain tumor classification from MRI images. In *IFIP advances in information and communication technology* (Vol. 583, pp. 189–200), IFIP.
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*
13. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations (ICLR)*.
14. Geetha, R., & Thilagam, T. (2021). A review on the effectiveness of machine learning and deep learning algorithms for cyber security. *Archives of Computational Methods in Engineering, 28*(4), 2861–2879.
15. Li, L., Yan, J., Wang, H., & Jin, Y. (2021). Anomaly detection of time series with smoothness-inducing sequential Variational auto-encoder. *IEEE Transactions on Neural Networks and Learning Systems, 32*, 1177–1191.
16. Brain Tumor Dataset. (n.d.). https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection
17. Raja, J., Shanmugam, P., & Pitchai, R. (2021). An automated early detection of glaucoma using support vector machine based visual geometry group 19 (VGG-19) convolutional neural network. *Wireless Personal Communications, 118*(1), 523–534.
18. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture, 161*, 272–279.
19. Vasan, D., et al. (2020). IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. *Computer Networks, 171*, 107138.
20. Singh, A., Bansal, A., Chauhan, N., Sahu, S. P., & Dewangan, D. K. (2021). Image generation using GAN and its classification using SVM and CNN. In *Proceedings of emerging trends and technologies on intelligent systems* (pp. 89–100).
21. Pathari, S., & Rahul, U. (2020). Automatic detection of COVID-19 and Pneumonia from chest X-Ray using transfer learning. *medRxiv.*
22. Fidon, L., Ourselin, S., & Vercauteren, T. (2021). Generalized Wasserstein Dice Score, distributionally robust deep learning, and ranger for brain tumor segmentation: BraTS 2020 challenge. In *Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries* (pp. 200–214).
23. Bajić, F., Orel, O., & Habijan, M. (2022). A multi-purpose shallow convolutional neural network for chart images. *Sensors, 22*, 7695.
24. Lei, S., Shi, Z., & Zou, Z. (2020). Coupled adversarial training for remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing, 58*, 3633–3643.
25. Liu, D., Liu, Y., & Dong, L. (2019). G-ResNet: Improved ResNet for brain tumor classification. In *Neural information processing* (pp. 535–545).

26. Chen, Z., Zhu, Y., Zhao, C., Hu, G., Zeng, W., Wang, J., & Tang, M. (2021). DPT: Deformable patch-based transformer for visual recognition. In *Proceedings of the 29th ACM international conference on multimedia*.
27. Hossain, T., Shishir, F. S., Ashraf, M., Al Nasim, M. A., & Muhammad Shah, F. (2019). Brain tumor detection using convolutional neural network. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*.
28. Kora, P., Mohammed, S., Surya Teja, M. J., Usha Kumari, C., Swaraja, K., & Meenakshi, K. (2021). Brain tumor detection with transfer learning. In *2021 Fifth international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*.
29. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics, 21*, 1–13.
30. Hashemzehi, R., Mahdavi, S. J. S., Kheirabadi, M., & Kamel, S. R. (2020). Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE. *Biocybernetics and Biomedical Engineering, 40*, 1225–1232.
31. Wongpakaran, N., et al. (2019). A comparison of Cohen's kappa and Gwet's AC1 when calculating inter-rater reliability coefficients: A study conducted with personality disorder samples. *BMC Medical Research Methodology, 13*(1), 1–7.
32. Deepak, S., & Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. *Computers in Biology and Medicine, 111*, 103345.
33. Majib, M. S., Rahman, M. M., Sazzad, T. S., Khan, N. I., & Dey, S. K. (2021). Vgg-scnet: A vgg net-based deep learning framework for brain tumor detection on MRI images. *IEEE Access, 9*, 116942–116952.
34. Toğaçar, M., Ergen, B., & Cömert, Z. (2020). BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model. *Medical Hypotheses, 134*, 109531.
35. Zaw, H. T., Maneerat, N., & Win, K. Y. (2019). Brain tumor detection based on Naïve Bayes classification. In *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*.
36. Hossain, T., Shishir, F. S., Ashraf, M., Al Nasim, M. A., & Shah, F. M. (2019). Brain tumor detection using convolutional neural network. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*.
37. Younis, A., et al. (2022). Brain tumor analysis using deep learning and VGG-16 ensemble learning approaches. *Applied Sciences, 12*(14), 7282.
38. Amin, J., Anjum, M. A., Sharif, M., Jabeen, S., Kadry, S., & Moreno Ger, P. (2022). A new model for brain tumor detection using ensemble transfer learning and quantum variational classifier. In *Computational intelligence and neuroscience* (pp. 1–13).

Combining Super-Resolution GAN and DC GAN for Enhancing Medical Image Generation: A Study on Improving CNN Model Performance

Mahesh Vasamsetti, Poojita Kaja, Srujan Putta, and Rupesh Kumar

1 Introduction

One of the common known cancers is skin cancer [5], and effective treatment depends on early detection. Skin cancer can have a terrible impact on one's health and well- being. It is a severe and sometimes fatal condition. One such cancer is skin cancer [6] which affects the skin, the biggest organ in the body. It attacks when aberrant skin cells multiply and grow out of control, frequently developing a cancerous tumor. Skin cancer is typical cancer, and its frequency has recently increased. Cancer can be caused by various factors, including sun exposure and ultraviolet (UV) radiation. Skin cancer can take many different forms [7], each with its own specific characteristics, such as melanoma, squamous cell carcinoma, basal cell carcinoma [8].

Traditional diagnostic techniques, however, are frequently pricy and time consuming. So, in this project, we use one of the neural networks called GANs (Generative Adversarial Networks) to present a viable remedy for this issue [9], a cost-effective way to detect skin cancer quickly and accurately.

GANs are composed of two networks: the generator and the discriminator. The generator network produces data fed into the discriminator network, which evaluates whether or not the generated data is fake or real. This back-and-forth between the two networks allows GANs to learn how to develop increasingly realistic data over time. Here we use two types of GANs: DC GAN and super-resolution GAN. A particular generative model called DCGAN (Deep Convolutional Generative Adversarial Network) employs deep learning methods to produce new images from a given dataset.

M. Vasamsetti (✉) · P. Kaja · S. Putta · R. Kumar
ECE Department, SRM University, Amaravati, Andhra Pradesh, India
e-mail: rupesh.k@srmap.edu.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_11

It works by training two neural networks, the discriminator and the generator, against each other in an adversarial process. The generator creates new images while the discriminator evaluates them based on their proximity to the original dataset. This process allows DCGANs to generate realistic images and can be used in applications like image-to-image translation or image synthesis. Super-resolution GAN, or SRGAN, is a type of GAN specifically designed to enhance the resolution of images. SRGANS use deep neural networks to generate high-resolution images from low-resolution images, which can be especially useful for works such as upscaling low-resolution images for high-resolution images [10].

GANs have various applications in various fields. Here are a few examples:

1. Image and video generation: GANs help produce new images and videos similar to real ones. GANs have applications in fields such as art, entertainment, and advertising.
2. Text-to-image synthesis: GANs can generate images from textual descriptions. GANs have applications in fields like e-commerce, where product images can be generated automatically based on text descriptions.
3. Music generation: GANs can generate new music similar to existing themes. This has applications in fields such as music production and composition.
4. Data augmentation: GANs are used to produce new information and can be used to generate new information, which is data that can be used to augment existing datasets [37]. GANs have applications in fields such as computer vision, where larger datasets can improve the performance of machine-learning models.
5. Simulation and modelling: GANs are used to produce synthetic data that can be used to train and test simulation and modelling systems [38]. GANs have applications in fields such as robotics, where simulated environments can be used to test and improve robot performance.
6. Anomaly detection: GANs can detect anomalies in data by comparing accurate data to generated data. This has applications in fields such as fraud detection and cybersecurity. These are some applications of GANs in various fields.

2 Related Work

In this chapter, the main focus was on the usage of GAN in image processing fields like image super-resolution, image-to-image translation, and cartoon generation. In this chapter, they discuss the challenges faced by GAN and their approach to overcoming them. This chapter also promises future enhancements in image processing and GAN and gives scope for tasks like face reconstruction [29]. The chance of survival for skin cancer patients is high when it is detected in early stages but in most cases, it is undetected until advanced stages. So, this chapter addresses the problem of early detection of cancer using GAN image processing. Collecting datasets is the major issue for medical image processing, and in this chapter, they

compensate for it by generating synthetic images from the given dataset and adding them to the existing dataset thereby increasing the accuracy of the detection [11].

Interstitial lung disease is a chronic lung disease and detection also becomes tough because of the diversity of causes and the irregular patterns of the lung tissues involved, so it becomes a big problem to detect the issue. In this chapter, the HRCT images will be correctly classified using the universal datasets available regarding lung diseases which help them easily diagnose the disease. It also uses deep learning techniques which may also help them cure several lung diseases [12].

Even though medical imaging is essential in various clinical applications, it does contain a few limitations like cost and radiation dose. In this chapter, an FCN (Fully Conventional Network) is trained to generate a target image given a source image. To get more realistic images, the FCN will implement an adversarial learning strategy and application of auto-context model to train the image gradient difference-based loss function to get less blurry images [13].

This chapter uses GANs to synthesize cells imaged by fluorescence microscopy. These generate new models with casual dependencies between image channels and can generate multi-channel images. They generate two different techniques and compare them to a sensible baseline. At the end interpolating across the latent space allowing us to predict temporal evolution from static images [14].

The chance of survival for brain tumor patients is high when it is detected in early stages but in most cases, it is undetected until advanced stages. This chapter proposes a method for segmenting brain tumors in MRI images using a GAN. The GAN is trained on both normal and tumor images to generate new tumor images. The key difference of this chapter is that it focuses on tumor segmentation rather than image generation [15].

This chapter focuses on skin cancer classification using ECOC SVM (Support Vector Machine) and deep convolutional neural networks on RGB images collected from the Internet. The pretrained AlexNet model is used to extract features, and a proposed algorithm achieves high accuracy, sensitivity, and specificity on a total of 3753 images, including four types of skin cancers. The results show maximum accuracy for actinic keratosis, high sensitivity for squamous cell carcinoma, and high specificity for squamous cell carcinoma. Still, some measures fall slightly below the maximum for basal cell carcinoma, melanoma, and squamous cell carcinoma [16].

The chapter demonstrates using a deep convolutional neural network to classify skin lesions. The CNN is trained on a dataset of 129,450 clinical images and tested against 21 board-certified dermatologists on biopsy-proven clinical images for two binary classification tasks. The CNN performs comparably to dermatologists, showing potential for extending the reach of dermatologists beyond the clinic via mobile devices [17].

This chapter comprehensively reviews GANs in medical image analysis, including image generation, segmentation, registration, and synthesis. The chapter's authors use a GAN to learn the distribution of regular data and use it to detect anomalies. They also introduce a novel marker discovery approach that identifies regions of the image that are most responsible for anomaly detection. The authors

show that their method can be applied to several domains, including medical imaging, and can accurately detect anomalies [39].

The chapter's authors use a GAN to generate synthetic medical images similar to authentic medical images. They also introduce a novel loss function that encourages the generated images to have similar statistical properties as the authentic images. The authors show that their method can be applied to several medical imaging modalities, including MRI and CT, and can generate high-quality images useful for training machine learning models [40].

The chapter's authors use dual GANs to generate synthetic medical images similar to authentic medical images. They also introduce a novel loss function that encourages the generated images to have similar statistical properties as the authentic images. The authors show that their method can be applied to several medical imaging modalities, including MRI and CT, and can generate high-quality images useful for training machine learning models. Compared to other methods, the dual GAN approach generates more diverse images with higher visual quality [41].

The authors summarize the types of GANs used for medical image analysis, internal GANs, cycle-consistent GANs, and adversarial autoencoders. They also discuss the applications of GANs in medical image analysis, such as generating synthetic images for data augmentation, segmenting medical images, and registering medical images. The authors highlight the potential of GANs to improve the accuracy and efficiency of medical image analysis algorithms. However, they also note that several problems must be overcome, like better evaluation metrics and more diverse and representative datasets [42].

The chapter's authors propose a context-aware generative adversarial network (CA- GAN) that can synthesize medical images while preserving the contextual information of the images. They introduce a novel loss function that encourages the generated images to have statistical properties similar to the authentic images while preserving the contextual information. They demonstrate their method's effectiveness using several medical imaging modalities, including MRI and CT. The authors show that their way can generate more diverse images with higher visual quality than other methods. They also show that their method can be used for data augmentation, improving the performance of medical image analysis algorithms. Finally, they discuss the limitations of their method and provide recommendations for future research [43].

3 Methodology

3.1 Dataset

In our study, we used the International Skin Imaging Collaboration's (ISIC) skin cancer dataset, which was made available by Kaggle (ISIC) [18]. The collection includes 2357 photos of oncological disorders, both benign and malignant, that

were grouped according to the ISIC categorization. Actinic keratosis, nevus, basal cell carcinoma, melanoma, dermatofibroma, squamous cell carcinoma, pigmented benign keratosis, and seborrheic keratosis are among the different skin conditions represented in the dataset. We have used four classes from this dataset which are basal cell carcinoma, melanoma, squamous cell carcinoma, and pigmented benign keratosis; three of them are cancerous, and the fourth is noncancerous. Figure 1 contains the images of each class from the dataset.

Melanoma: Skin cancer called melanoma develops in the cells that make melanin, the pigment that gives skin, hair, and eyes their color [19]. The face, neck, arms, and legs are among the body parts most frequently affected by melanoma, though it can develop anywhere on the body [20]. Additionally, melanoma can develop in the eyes and other body regions with pigment-producing cells.

Basal Cell Carcinoma: Basal cells, which make up the lowest layer of the epidermis (the skin's outer layer), are where basal cell carcinoma (BCC), a kind of skin cancer, occurs. The most frequent type of skin cancer, BCC is typically brought on by UV radiation from the sun or tanning beds. BCC typically manifests as a tiny, glossy lump or nodule on the skin that frequently has blood vessels that are visible. It could also resemble a white, waxy scar or a red, scaly spot [21].

Squamous Cell Carcinoma: Squamous cells, which are the flat, thin cells that make up the skin's outer layer, are where squamous cell carcinoma (SCC), a specific kind of skin cancer, originates [22]. SCC can form on skin areas that have been damaged or exposed to radiation, but it is typically brought on by exposure to ultraviolet (UV) radiation from the sun or tanning salons [23]. SCC often manifests as a sore or patch that is red, scaly, and may bleed or crust over. Additionally, it could resemble a wart or a raised, scaly lump [24].

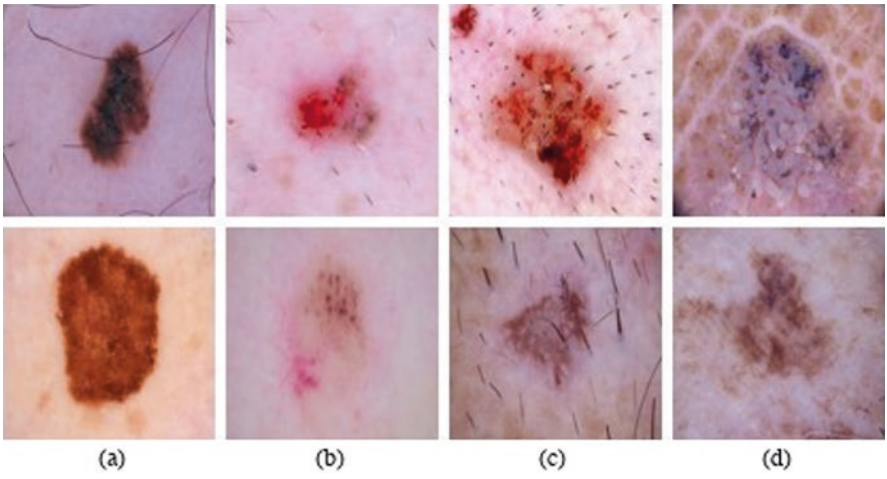


Fig. 1 (**a**) Melanoma, (**b**) basal cell carcinoma, (**c**) squamous cell carcinoma, and (**d**) pigmented benign keratosis

Pigmented Benign Keratosis: Seborrheic keratosis, also known as pigmented benign keratosis (PBK), is a common noncancerous skin growth that often manifests as a raised brown or black lesion on the skin. The majority of people with PBK are middle-aged or older, and it frequently runs in families [25]. Although PBK typically causes no symptoms and is painless, it can be ugly and may be mistaken for melanoma, a more dangerous form of skin cancer [26]. A biopsy may be carried out to confirm the diagnosis of PBK, which is typically made based on how the condition appears.

3.2 Algorithm

Our entire process consists of three phases. The first phase is about balancing the dataset. Initially, basal cell carcinoma contains 376 images, squamous cell carcinoma contains 181 images, and melanoma contains 438 images. We train a DCGAN model on basal cell carcinoma and squamous cell carcinoma to generate synthetic images and increase each class's size to 500 images. The second phase involves using SRGAN to upscale the images generated by the DCGAN model which are 64×64 resolution to 255×255 resolution. This technique is used to increase the details of the images. Finally, in the third phase, we train a ResNet18 model on both the original dataset and the synthetic images generated by the DCGAN model. Figure 2 shows the entire flow chart of this process.

3.2.1 DCGAN

Over the initial GAN [27], the Deep Convolutional Generative Adversarial Networks (DCGAN) [28] provide a significant improvement. Using DCGAN, it is now easier to produce high-quality images and achieve stability during the training phase. Training and generation are the two steps of the DCGAN synthetic image-generating process. The generator creates images during training by taking samples from an N-dimensional normal distribution and applying sequential up-sampling operations

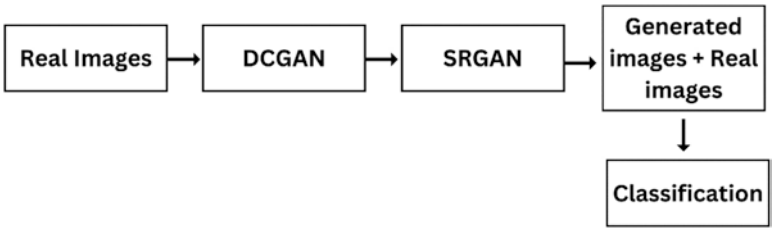


Fig. 2 Flow chart

to a random input noise vector. Contrarily, the discriminator seeks to distinguish between the pictures produced by the generator and those in the training set [29].

BatchNorm [29] to normalize the extracted feature scale and Leaky ReLU [30] to prevent vanishing gradients are two essential elements that DCGAN includes. Convolutional stride takes the role of all max pooling in DCGAN, and transposed convolution is used for up-sampling. Fully linked layers are eliminated, and batch normalization is used in their place. ReLU is used in the generator whereas Leaky ReLU is used in the discriminator, with the exception of the output, which utilizes tanh.

3.2.2 SRGAN

SRGAN aims to produce a high-resolution image from a low-resolution image. A generator network is used in the SRGAN [31], and it uses residual blocks to preserve data from earlier layers and enable the network to make adaptive selections from a broader range of characteristics. With SRGAN, we feed the low-resolution image as input to the generator network as opposed to typical GANs, where random noise is supplied as the generator input. The discriminator network is relatively conventional and functions similarly to how a discriminator would result in a typical GAN. The perceptual loss function is what makes SRGANs unique. SRGANs employ the perceptual/content loss function to get where they are going while the discriminator and generator are trained using the GAN architecture. The perceptual loss function is intended to assist the SRGAN in building a loss function that accomplishes its objective by identifying the perceptually essential properties. So, in addition to the content loss, the adversarial loss also contributes to the adjustment of the weights [32]. The output of a previously trained VGG (Visual Geometry Group) network is compared pixel-by-pixel to describe the content loss as a VGG loss. Only when the input images are comparable will the actual VGG output and the fake VGG output be similar [32]. The idea is that pixel-by-pixel comparison will enhance the primary goal of obtaining super-resolution. The combined effects of the content loss and the GAN loss are favorable. The resulting images with super-resolution are evident and accurate representations of their high-resolution counterparts. The perceptual loss function minimizes information loss during the image upscaling process, producing moving and identical images to high-resolution images (Figs. 3 and 4).

3.2.3 ResNet18

ResNet-18 is a deep convolutional neural network with 18 layers. It is a member of the ResNet network family, renowned for its intricate structure and superior performance on image recognition tasks single convolutional layer. It is the first layer in the ResNet-18 design, with 18 layers (Fig. 5).

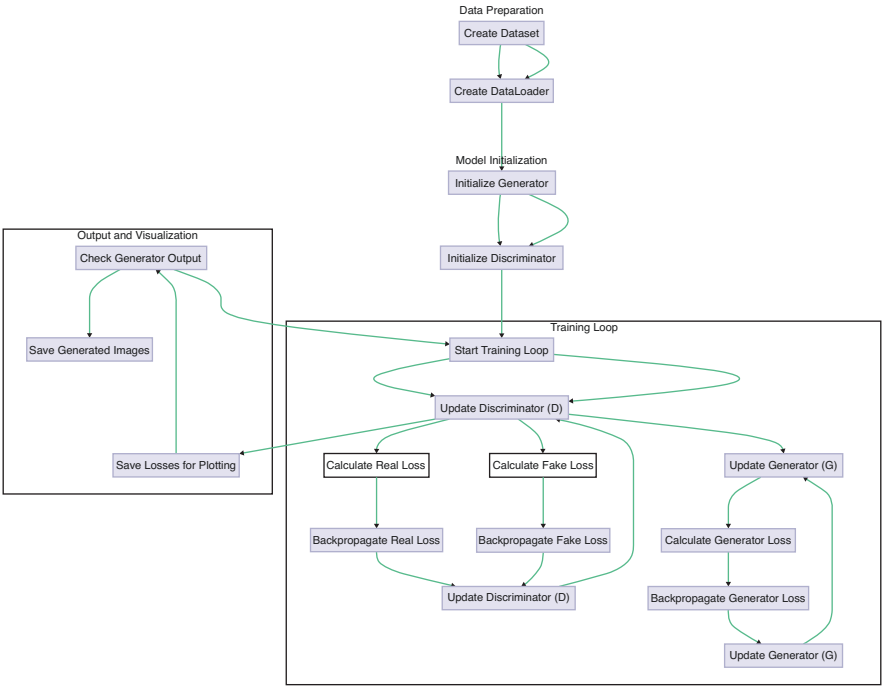


Fig. 3 Flow chart of DCGAN

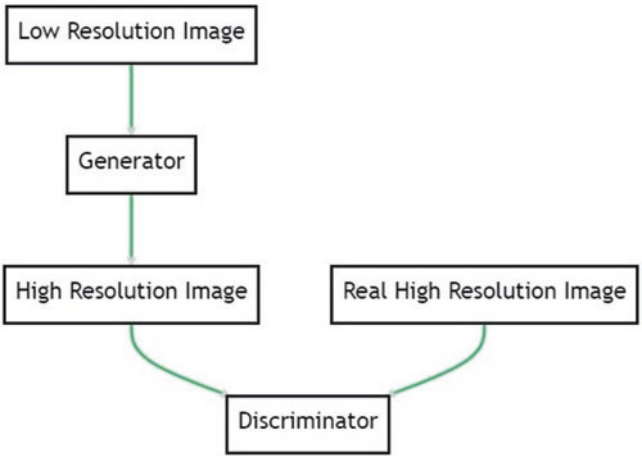


Fig. 4 Flow chart of SRGAN

In total [33], there are then four blocks of layers. The size of the input picture is cut in half by the first convolutional layer, which has a kernel size of 7×7 and a stride of 2. The image size is further decreased by passing the output of the first

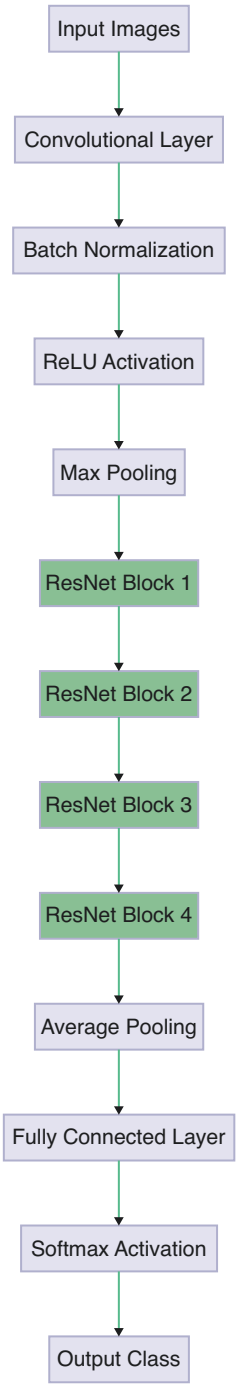


Fig. 5 Flow chart of ResNet18

convolutional layer through a max pooling layer with a kernel size of 3 × 3 and a stride of 2 [34]. In ResNet-18, each layer block comprises two or three convolutional layers followed by a shortcut link that adds the output of the convolutional layers to the block's input. This shortcut link helps avoid the vanishing gradients problem in deep neural networks [33].

Due to the shortcut link, the network may also learn residual functions, hence the name "ResNet." ResNet-18's last layer block has three convolutional layers instead of the first two-layer blocks' two each. The total quantity of filters in each one increases from 64 in the first block to 512 in the last block as we move further into the network. A global average pooling layer in ResNet-18 receives the output of the previous convolutional layer and averages the output features across spatial dimensions. The final output classification probabilities are produced by processing the result of the global intermediate pooling layer via a fully linked layer with an activation function based on softmax [35].

3.3 Results

3.3.1 DCGAN

We have trained DCGAN with learning rate of 0.001 on melanoma class which contains 438 images for 700 epochs, basal cell carcinoma contains 376 images for 700 epochs, and squamous cell carcinoma contains 181 images for 700 epochs. After training the DCGAN for 700 epochs on melanoma images, the generated images are displayed in Fig. 6.

Discriminator and generator loss during training on melanoma pictures which were trained for 700 epochs are displayed in Fig. 7.

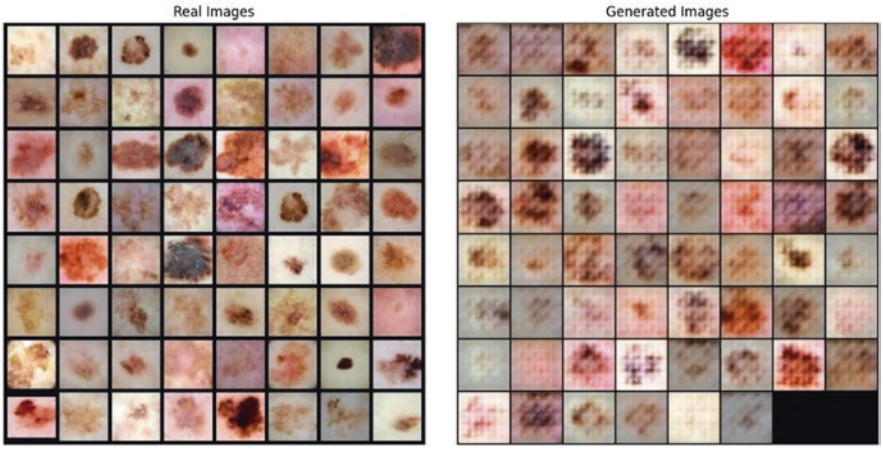


Fig. 6 Real images (left side), generated images (right side)

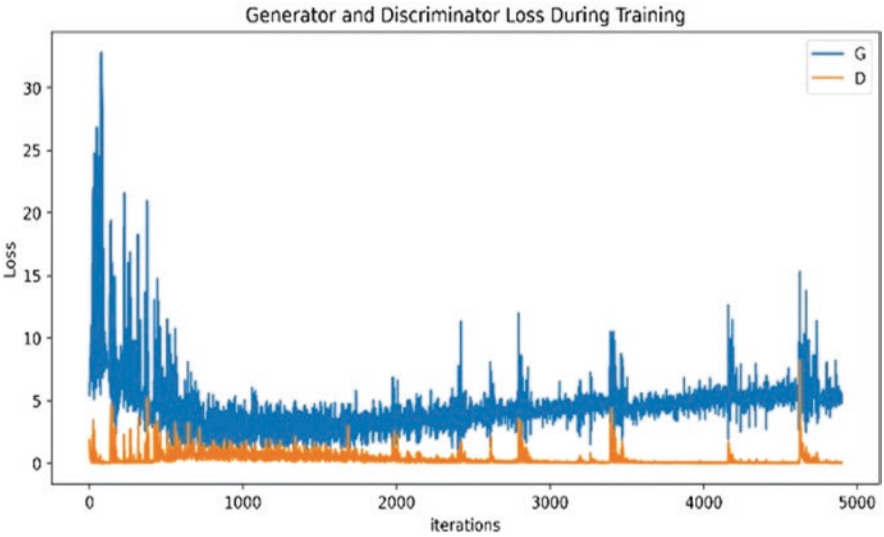


Fig. 7 Generator and discriminator loss during training

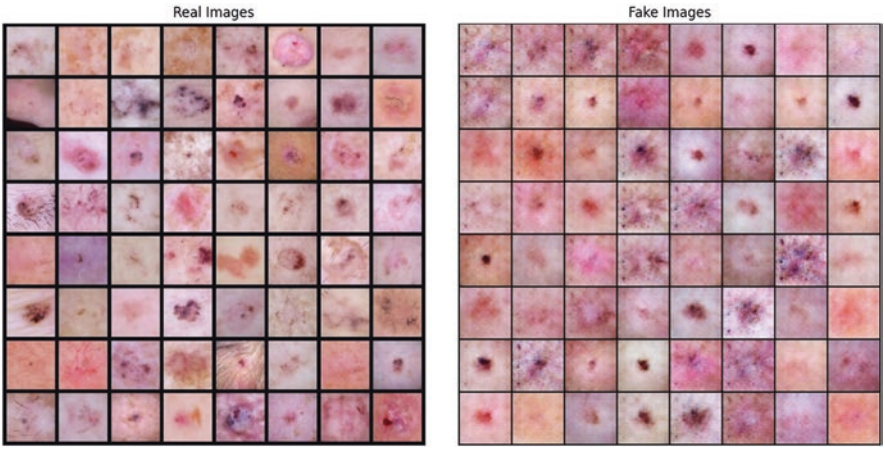


Fig. 8 Real images (left side), generated images (right side)

After the training process was completed, Fig. 6 displayed a total of 62 synthetically generated images that resembled skin lesions. After training the DCGAN for 700 epochs on basal cell carcinoma images, the generated images are displayed in Fig. 8.

Discriminator and generator loss during training on basal cell carcinoma images which were trained for 700 epochs are displayed in Fig. 9.

After the training process was completed, Fig. 8 displayed a total of 124 synthetically generated images that resembled skin lesions. After training the DCGAN

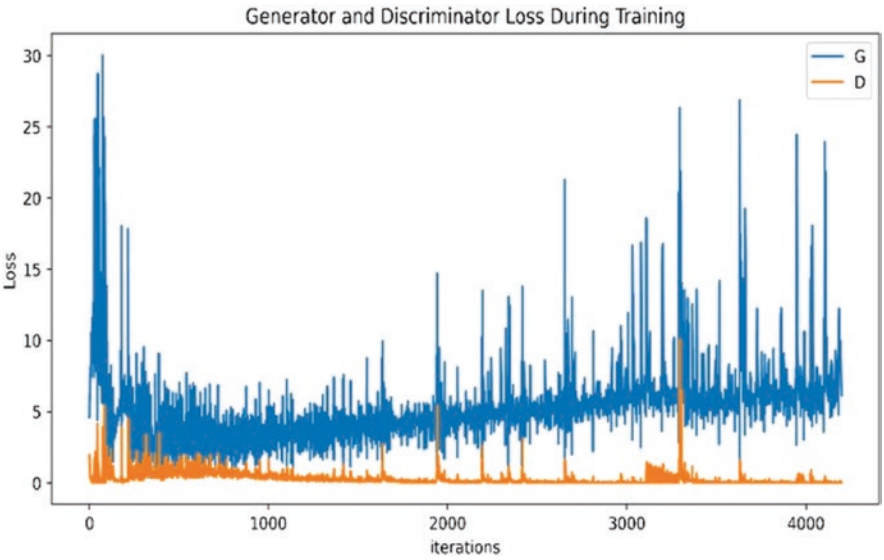


Fig. 9 Discriminator and generator loss during training

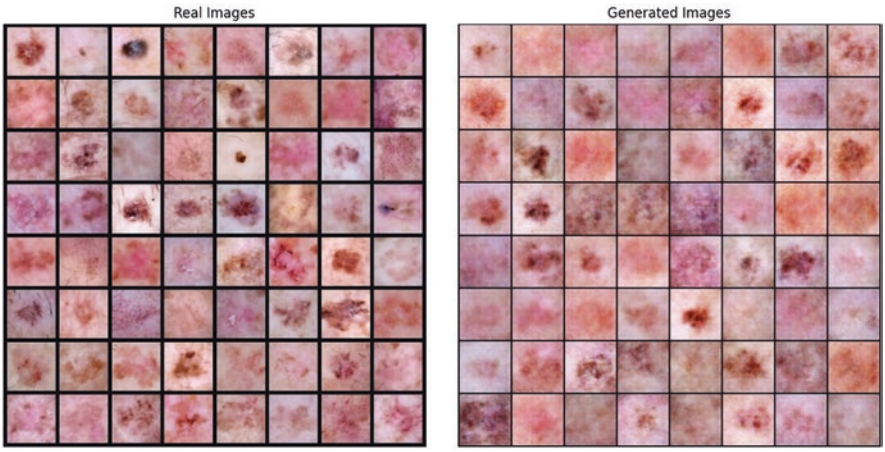


Fig. 10 Real images (left side), generated images (right side)

for 700 epochs on squamous cell carcinoma images, the generated images are displayed in Fig. 10.

Discriminator and generator loss during training on squamous cell carcinoma images which were trained for 700 epochs are displayed in Fig. 11.

After the training process was completed, Fig. 10 displayed a total of 319 synthetically generated images that resembled skin lesions. However, these images did not possess enough realism to deceive a dermatologist. Although they appeared

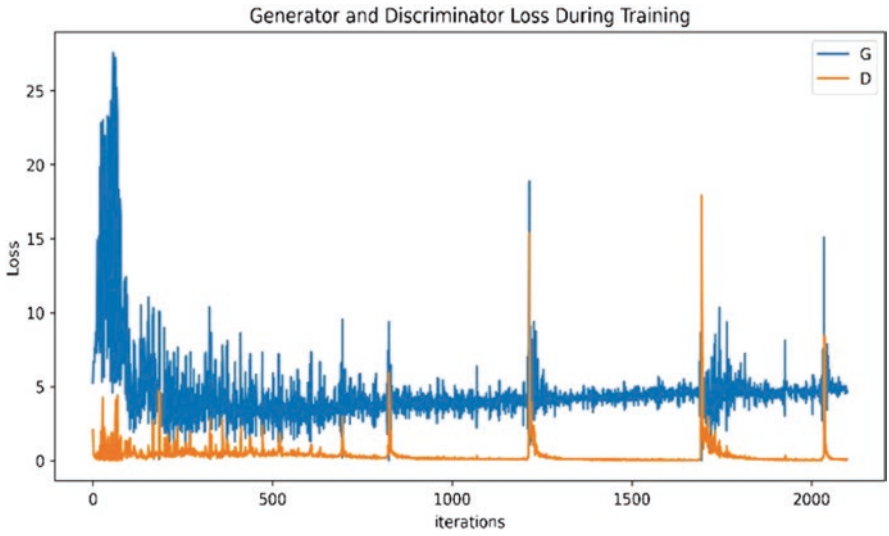


Fig. 11 Discriminator and generator loss during training

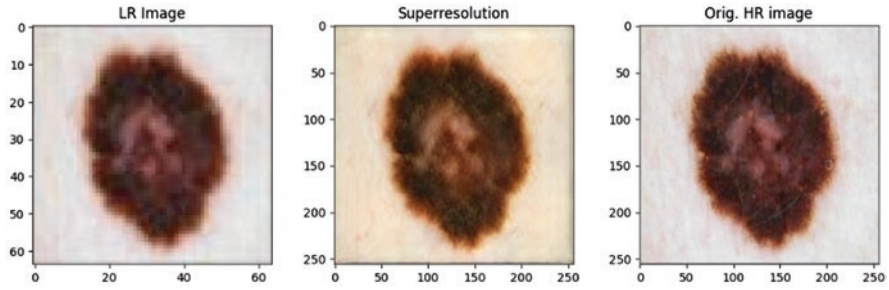


Fig. 12 Melanoma

diverse in nature, capturing the diversity of the training set used by the discriminator [36], several imperfections were observed. One notable issue was the presence of a noisy periodic pattern, which was visible in an 8x8 grid of blocks across the image. Additionally, other artefacts were also visible in the images, detracting from their overall quality.

3.3.2 SRGAN

The SRGAN was utilized to upscale low-resolution images generated by the DCGAN. Specifically, the DCGAN produced synthetic images of size 64 × 64. when trained on the original dataset. These images were then fed into the SRGAN

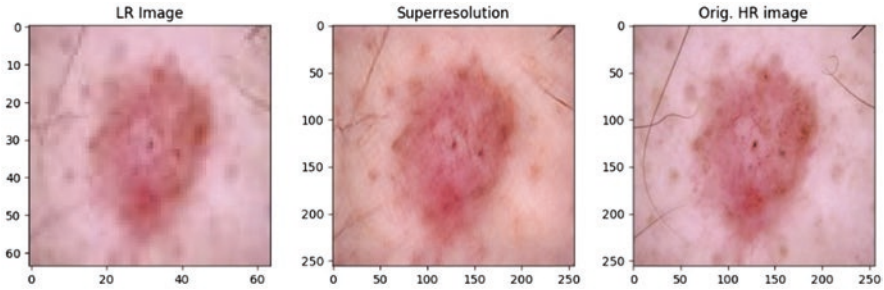


Fig. 13 Basal cell carcinoma

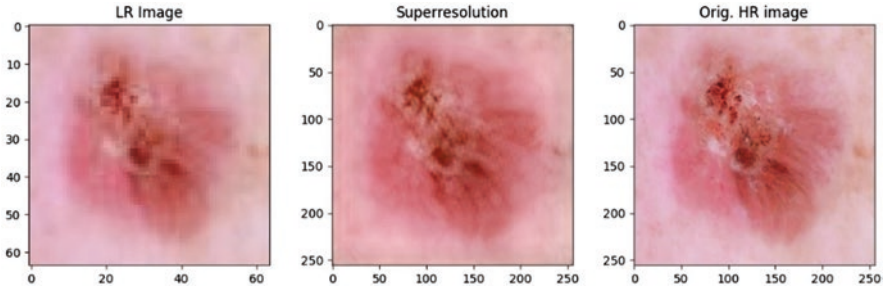


Fig. 14 Squamous cell carcinoma

for super-resolution upscaling. The SRGAN upscaled image of the Melanoma class image was shown in Fig. 11 when trained on original dataset [9] (Fig. 12).

The upscaled image of the basal cell carcinoma class image is displayed in Fig. 13 after SRGAN was trained on the original dataset [9].

The upscaled image of the squamous cell carcinoma class image is displayed in Fig. 14 after SRGAN was trained on the original dataset [9].

3.3.3 ResNet18

Two ResNet-18 models were trained to classify skin cancer. The first model was trained using the original dataset for four epochs and achieved an accuracy of 73%. This model was trained using the standard approach of feeding the original dataset into the network during training. The second model was also trained for four epochs, but in addition to the original dataset, it was also trained with fake images generated by another DCGAN. The original dataset was updated with these simulated photos in order to speed up the model (Table 1).

This approach of using fake images to augment the dataset is known as data augmentation, and it is the most used technique in machine learning. The idea is to produce additional examples of training which could help in learning more robust characteristics for the model and minimize overfitting.

Table 1 Results

Architecture	Original Dataset (Accuracy)	Original + Generated images (Accuracy)
Resnet18	82%	85%

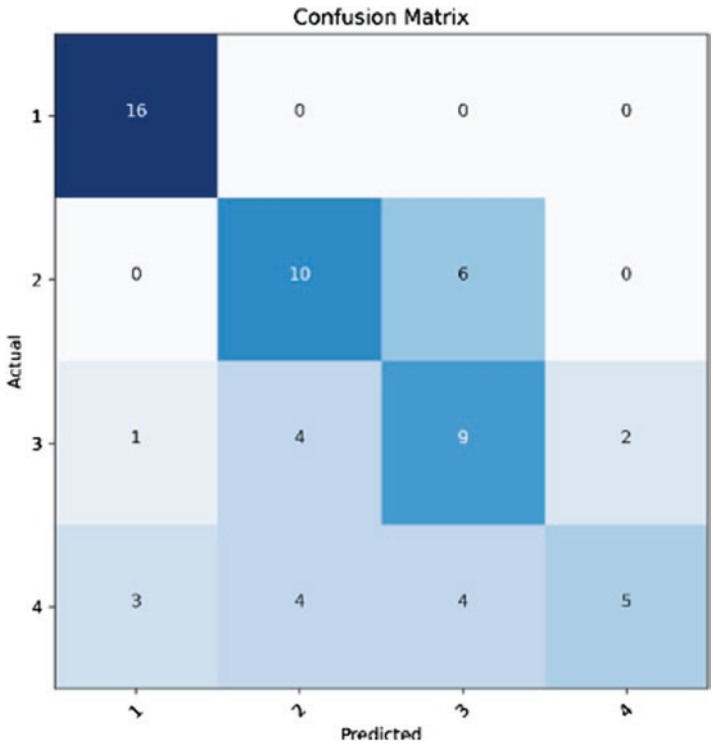


Fig. 15 Confusion Matrix of ResNet18 trained on the original dataset

The second model's accuracy was evaluated after it had been trained using the new dataset and compared to the first model's accuracy. Results indicate that the second model outperformed the first model in accuracy. This shows that the model's performance was enhanced by introducing fake photos for data augmentation.

This study evaluated the performance of a ResNet18 model trained on an original dataset. We compared it to a ResNet18 model trained on the original dataset and generated images. The confusion matrix of ResNet18 trained on the original dataset is shown in Fig. 15.

We evaluated the performance of both models on 16 unseen images for each class. Adding created images to the primary dataset improved the model's generalization. The confusion matrix of ResNet18 trained on the original dataset and generated images are shown in Fig. 16. Confusion matrix 1,2,3,4 indicates the squamous cell carcinoma class, pigmented benign keratosis class, melanoma class, and basal cell carcinoma class. As it is commonly known, using generated images with

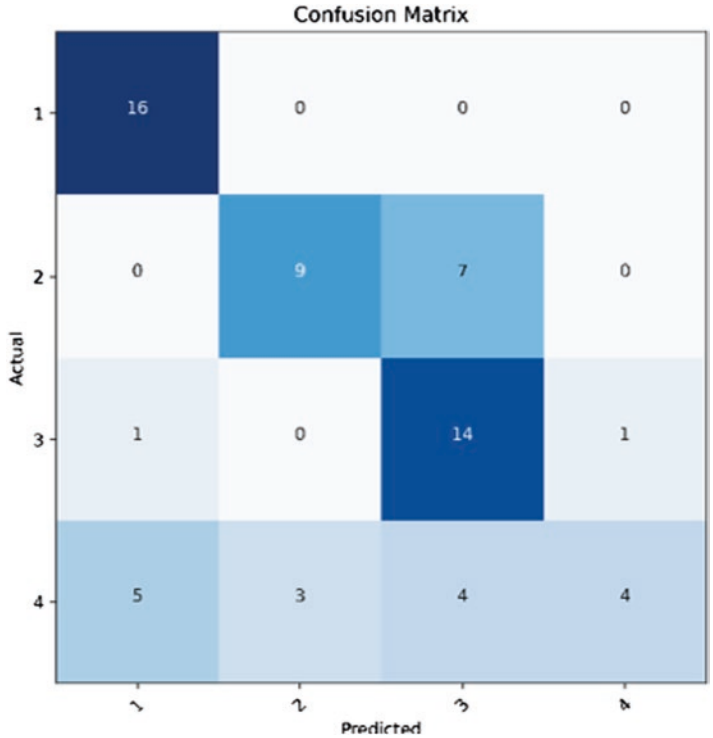


Fig. 16 Confusion matrix of ResNet18 trained on generated images and original dataset

original images for training can enhance the model's ability to generalize; our work indicates that applying image super-resolution techniques to the generated images as a substitute for traditional image processing techniques on the original images leads to an increase in image details and further improving the generalization of the model when trained with these images.

3.4 Conclusion

The proposed approach for generating synthetic skin lesion images involved using a combination of different deep-learning techniques. Firstly, a DCGAN was trained on the original dataset to generate 64 × 64 images of skin lesions. These images were then upscaled using an SRGAN to improve their resolution. Here, it is essential that you keep mindful that the images from DCGAN are not perfectly similar to the original images in terms of quality. Despite these imperfections, the generated images demonstrated that the proposed solution successfully generated synthetic skin lesions. However, the flaws highlight the ongoing challenges in generating highly realistic images using GANs. This was solved using SRGAN. SRGANs have

delivered outstanding outcomes in image upscaling by including the perceptual loss function in the GAN architecture, creating new possibilities for using low-resolution images in various applications. SRGANs are anticipated to advance in sophistication and power over the next few years with sustained research and development, revolutionizing how humans see and interpret images. SRGANs have several uses in many industries, such as satellite imaging, surveillance, and medical imaging. SRGANs, for instance, can help create high-resolution images of medical scans, facilitating more precise treatment and patient diagnosis in the health industry. SRGANs can be used in the surveillance industry to improve low-resolution security camera video, making recognizing criminals and solving crimes simpler. The proposed approach shows promise in generating synthetic skin lesion images that can enhance the performance of skin lesion classifiers. However, extra research must be done to address the limitations in the amount of the artificial pictures and explore the potential of this approach in other medical imaging applications.

References

1. Skin cancer (Including Melanoma)—Patient version. (n.d.). National Cancer Institute. https://www.cancer.gov/types/skin#:~:text=Skin%20cancer%20is%20the%20most,other%20parts%20of%20the%20body
2. Sunstation USA. *What is melanoma?* https://www.sunstationusa.com/single-post/2017/05/25/what-is-melanoma
3. Shao, C., Dai, W., Li, H., Tang, W., Jia, S., Wu, X., & Luo, Y. (2017). The relationship between RASSF1A gene promoter methylation and the susceptibility and prognosis of melanoma: A meta-analysis and bioinformatics. *PLoS One, 12*(2), e0171676.
4. Skin cancer. (2006, December 31). WebMD. https://www.webmd.com/melanoma-skin-cancer/melanoma-guide/skin-cancer#1
5. Skin cancer - Symptoms and causes - Mayo Clinic. (2022, December 6). Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605
6. Mutepfe, F., Kalejahi, B. K., Meshgini, S., & Daneshvar, S. (2021). Generative adversarial network image synthesis method for skin lesion generation and classification. *Journal of Medical Signals and Sensors, 11*(4), 237. https://doi.org/10.4103/jmss.jmss_53_20
7. Health Perfecto. *Skin cancer symptoms, causes, risk factor, and treatment*. https://www.health-perfecto.com/skin-cancer/
8. Nima Skin Institute. *Melanoma archives*. https://nimaskininstitute.com/tag/melanoma/
9. Mohs Surgery MD. *Skin Cancer Surveillance Specialist – Chevy Chase, MD: Ali Hendi, MD: Skin Cancer Specialist*. https://www.mohssurgerymd.com/services/skin-cancer-surveillance
10. Chakraborty, D. (2022). Super-Resolution Generative Adversarial Networks (SRGAN). *PyImageSearch*. https://www.pyimagesearch.com/2022/06/06/super-resolution-generative-adversarial-networks-srgan/
11. Rashid, H., Tanveer, M. A., & Khan, H. A. (2019. Skin lesion classification using GAN-based data augmentation. In *2019 41st annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE.
12. Gao, Mingchen, et al. (2017). Holistic interstitial lung disease detection using deep convolutional neural networks: Multi-label learning and unordered pooling. *arXiv preprint arXiv:1701.05616*.
13. Nie, D., et al. (2018). Medical image synthesis with deep convolutional adversarial networks. *IEEE Transactions on Biomedical Engineering, 65*(12), 2720–2730.

14. Osokin, Anton, et al. (2017). GANs for biological image synthesis. In *Proceedings of the IEEE international conference on computer vision*.
15. Myronenko, Andriy. (2019). 3D MRI brain tumor segmentation using autoencoder regularization. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*. Springer International Publishing.
16. Dorj, U.-O., et al. (2018). The skin cancer classification using deep convolutional neural network. *Multimedia Tools and Applications, 77*, 9909–9924.
17. Esteva, A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature, 542*(7639), 115–118.
18. Kaggle. *Skin cancer ISIC dataset*. https://www.kaggle.com/datasets/nodoubttome/skin-cancer9-classesisic
19. Centers for Disease Control and Prevention. *What is skin cancer?* https://www.cdc.gov/cancer/skin/basic_info/what-is-skin-cancer.htm
20. Dallas Dermatology. *Skin cancer dermatologist – Skin cancer treatments*. https://www.dallasdermpartners.com/medical-dermatology-dallas/skin-cancers/
21. Mayo Clinic. *Basal cell carcinoma*. https://www.mayoclinic.org/diseases-conditions/basal-cell-carcinoma/symptoms-causes/syc-20354187
22. Mayo Clinic. *Squamous cell carcinoma of the skin*. https://www.mayoclinic.org/diseases-conditions/squamous-cell-carcinoma/symptoms-causes/syc-20352480
23. WebMD. *Squamous cell carcinoma: Symptoms, causes, diagnosis, treatment*. https://www.webmd.com/melanoma-skin-cancer/melanoma-guide/squamous-cell-carcinoma#1
24. Healthline. *Squamous cell cancer*. https://www.healthline.com/health/squamous-cell-skin-cancer
25. Medical News Today. *Seborrheic Keratosis: Symptoms, treatment, and causes*. https://www.medicalnewstoday.com/articles/266748
26. Verywell Health. *Seborrheic Keratosis: Symptoms, causes, diagnosis, and treatment*. https://www.verywellhealth.com/seborrheic-keratosis-1068732
27. Yale Medicine. *Seborrheic Keratosis*. https://www.yalemedicine.org/conditions/seborrheic-keratosis
28. NHS Inform. *Bile duct cancer*. https://www.nhsinform.scot/illnesses-and-conditions/cancer/cancer-types-in-adults/bile-duct-cancer-cholangiocarcinoma
29. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*.
30. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
31. Ioffe, S., & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
32. Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve the neural network, acoustic models. In *Proc. icml* (vol. 30, p. 3).
33. Demiray, B. Z., Sit, M., & Demir, I. (2021). D-SRGAN: DEM super-resolution with generative adversarial networks. *SN Computer Science, 2*, 1–11.
34. Ledig, C., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
35. Odusami, M., et al. (2021). Analysis of features of Alzheimer's disease: Detection of the early stage from functional brain changes in magnetic resonance images using a finetuned ResNet18 network. *Diagnostics, 11*(6), 1071.
36. Yu, X., & Wang, S. H. (2019). Abnormality diagnosis in mammograms by transfer learning based on ResNet18. *Fundamenta Informaticae, 168*(2–4), 219–230.

37. Zhang, Y., et al. (2020). A seven-layer convolutional neural network for chest CT-based COVID-19 diagnosis using stochastic pooling. *IEEE Sensors Journal, 22*(18), 17573–17582.
38. DataScienceCentral. *Synthetic image generation using GANs*. https://www.datasciencecentral.com/synthetic-image-generation-using-gans/
39. New-Impulse Media. *Generating passive income through AI Video creation: A comprehensive guide*. https://new-impulse.com/generating-passive-income-through-ai-video-creation-a-comprehensive-guide
40. Reason.town. *How deep learning can help with depth estimation*. https://reason.town/deep-learning-depth-estimation/.
41. Schlegl, T., et al. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information processing in medical imaging: 25th international conference, IPMI 2017,* Boone, NC, USA, June 25–30, 2017. Springer International Publishing.
42. Chen, Y., et al. (2022). Generative adversarial networks in medical image augmentation: A review. *Computers in Biology and Medicine, 144*, 105382.
43. Frid-Adar, M., et al. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing, 321*, 321–331.

GAN for Augmenting Cardiac MRI Segmentation

Pawan Whig, Pavika Sharma, Rahul Reddy Nadikattu, Ashima Bhatnagar Bhatia, and Yusuf Jibrin Alkali

1 Introduction

Cardiac magnetic resonance imaging (MRI) segmentation is an important task in medical imaging that involves separating different regions of the heart from each other in order to provide accurate diagnosis and treatment planning [1] as shown in Fig. 1. Deep-learning-based segmentation methods have shown promising results in recent years, but the performance of these methods is highly reliant on the availability and excellence of training data [2–8].

In this context, data augmentation techniques can be useful in increasing the diversity and quantity of training data, leading to better performance of deep learning-based segmentation methods. Generative adversarial networks (GANs) have shown great potential in data augmentation for medical image segmentation, but their application to cardiac MRI segmentation is relatively unexplored [9].

This chapter proposes the use of GANs for augmenting cardiac MRI segmentation data. Specifically, we develop a GAN-based data augmentation method that generates synthetic cardiac MRI images and evaluates the impact of this augmentation on the performance of a deep learning-based segmentation network. The proposed method aims to address the challenges of limited training data and

P. Whig (✉) · A. B. Bhatia
Vivekananda Institute of Professional Studies-TC, New Delhi, India

P. Sharma
Department of electronics and communication, BPIT, New Delhi, India
e-mail: pavikasharma@bpitindia.com

R. R. Nadikattu
Department of IT, Researcher, University of the Cumberland, Williamsburg, KY, USA

Y. J. Alkali
Department of IT, Federal Inland Revenue Service, Abuja, Nigeria

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_12

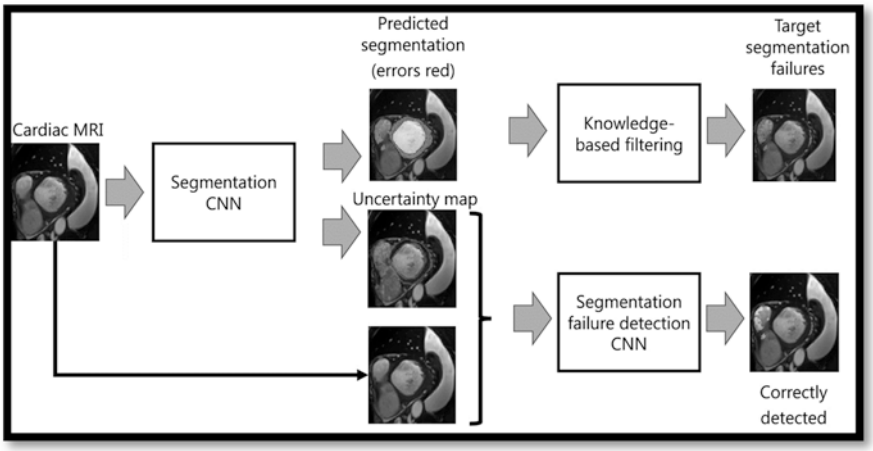


Fig. 1 Cardiac MRI segmentation

variability in cardiac anatomy, leading to improved segmentation accuracy and robustness [9].

GANs (Generative Adversarial Networks) have emerged as a powerful tool for medical image analysis, particularly in the realm of data augmentation techniques for medical image segmentation. Section 3 describes the methodology used in this study, including the dataset description, network architecture of GAN, and segmentation network architecture. Section 4 presents the results of our experiments, including a comparison of the performance of the segmentation network with and without GAN augmented data. Section 5 discusses the impact of our proposed method on cardiac MRI segmentation and provides directions for future research. Finally, Sect. 6 concludes the chapter.

1.1 Overview of Cardiac MRI Segmentation

Cardiac MRI segmentation involves the separation of different regions of the heart, such as the left ventricle, right ventricle, and myocardium, from each other in order to aid in the diagnosis and treatment planning of various cardiovascular diseases.

Accurate segmentation is critical for determining cardiac function and identifying abnormalities in the heart. However, the variability in cardiac anatomy and image quality makes segmentation a challenging task [2–8].

Deep learning-based segmentation, as shown in Fig. 2, often relies on advanced methods, such as convolutional neural networks (CNNs), to achieve high-precision delineation of regions of interest in medical images.

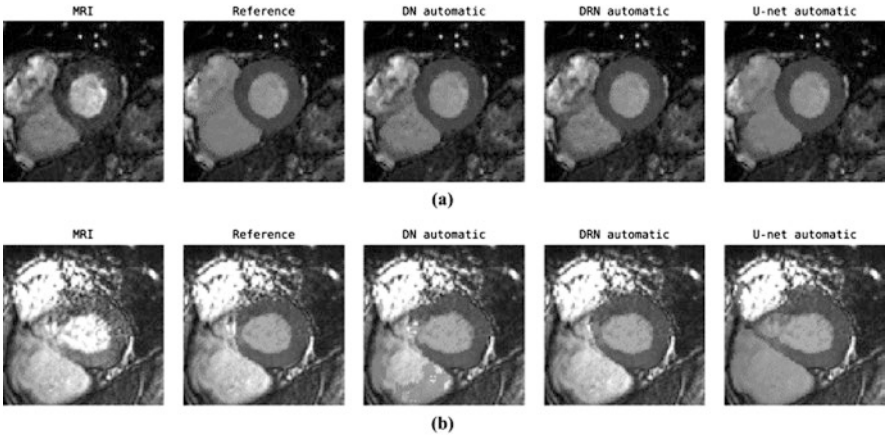


Fig. 2 Deep learning-based segmentation

1.2 Challenges in Cardiac MRI Segmentation

The heart is a complex organ with different shapes, sizes, and orientations, and this variability can make it difficult to develop a segmentation method that works well across all patients. Additionally, cardiac MRI images can suffer from various artifacts, such as noise, motion, and partial volume effects, which can further complicate segmentation [2–8].

Another challenge is the limited availability of annotated training data. Cardiac MRI images are typically acquired during the course of clinical exams, which are time-consuming and expensive. This can make it difficult to obtain a large amount of training data, particularly for rare or complex cardiac pathologies.

1.3 Introduction to GANs and Data Augmentation

GANs are a specific kind of deep learning model that is composed of two networks: a generator and a discriminator. The generator is responsible for generating artificial images, while the discriminator's role is to differentiate between real and artificial images. The generator learns to produce synthetic images that resemble the real images used during training, while the discriminator learns to recognize the disparities between real and synthetic images [2–8].

GANs have been used in various image synthesis and manipulation tasks, such as style transfer, image-to-image translation, and data augmentation. In the context of medical image segmentation, GANs can be used to generate synthetic images that mimic the variability in cardiac anatomy and image quality, leading to a more diverse and representative training set. Data augmentation techniques, such as

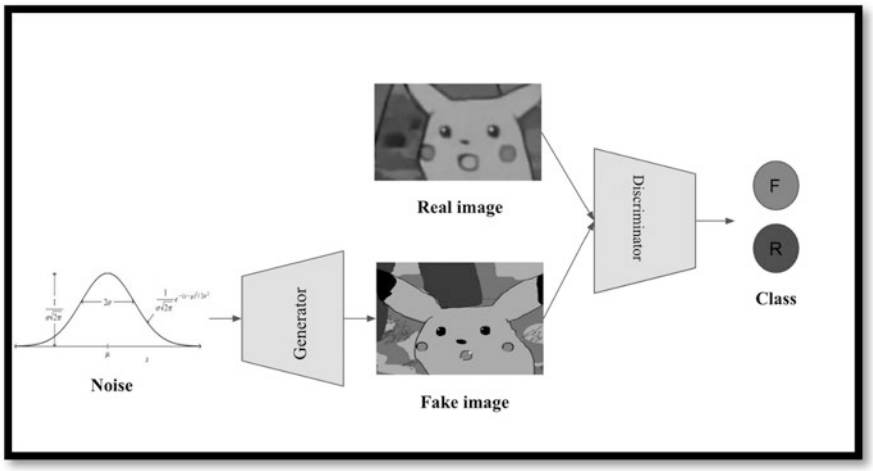


Fig. 3 GANs and data augmentation

GANs, can help to overcome the challenge of limited annotated training data, leading to improved segmentation accuracy and robustness [2–8] as shown in Fig. 3.

2 Literature Review of Cardiac MRI Segmentation

Segmentation of the heart in cardiac MRI is a challenging problem due to the variability in cardiac anatomy, image quality, and the presence of pathologies. Over the years, several approaches have been proposed to address this problem, ranging from traditional methods based on edge detection and thresholding to deep learning-based methods. Literature review of cardiac MRI segmentation is shown in Table 1.

One of the earliest approaches for cardiac MRI segmentation was the active contour method proposed in 1988. This method involves fitting a deformable contour to the edges of the heart in the image, and has been shown to be effective in segmenting the myocardium. However, this method is sensitive to initial conditions and can fail in the presence of noise or pathology.

In recent years, deep learning-based methods, such as convolutional neural networks (CNNs), have gained prominence in cardiac MRI segmentation. These methods have achieved state-of-the-art performance on various datasets and challenges [2–8].

To overcome challenges in cardiac MRI segmentation, several data augmentation techniques have been proposed, including the use of GANs. For example, a GAN-based data augmentation method was introduced, which generated synthetic images that captured the variability in cardiac anatomy and image quality. This approach demonstrated better performance compared to traditional data augmentation techniques and showed robustness to different imaging protocols and pathologies [10].

Table 1 Literature review of cardiac MRI segmentation

Approach	Description	Advantages	Limitations
Active contour	Deformable contour method that fits to the edges of the heart in the image	Effective in segmenting the myocardium	Sensitive to initial conditions, can fail in the presence of noise or pathology
CNN-based	Deep learning-based methods using convolutional neural networks	State-of-the-art performance, improved accuracy and robustness	Require a large amount of annotated training data, can be limited by generalization to different imaging protocols and pathologies
GAN-based data augmentation	Generative adversarial network-based techniques for augmenting training data	Captures variability in cardiac anatomy and image quality, better performance than traditional data augmentation techniques	Can be computationally expensive and time-consuming to generate synthetic images
Multi-modal fusion	Combining information from multiple imaging modalities, such as T1-weighted and T2-weighted MRI	Improved accuracy and robustness of the segmentation	Requires availability of multi-modal data, may increase computational complexity
Multi-task learning	Training the model to perform multiple related tasks, such as segmenting the myocardium and left ventricle	Improved accuracy and efficiency	Can be limited by the availability of annotated training data for multiple tasks
Attention mechanisms	Techniques that allow the model to selectively attend to different regions of the image	Improved accuracy and robustness of the segmentation	May increase computational complexity, can be limited by the availability of annotated training data for attention mechanisms

In addition to data augmentation, other techniques have been explored to improve the accuracy and robustness of cardiac MRI segmentation. These include multi-modal fusion, multi-task learning, and attention mechanisms. For instance, a multi-modal attention network was developed to combine information from both T1-weighted and T2-weighted MRI, enhancing the accuracy of the segmentation [11].

Overall, while significant progress has been made in cardiac MRI segmentation using deep learning-based methods, there are still challenges to be addressed, particularly in improving the generalizability of the models to different imaging protocols and pathologies. Data augmentation techniques, such as GANs, show promise in addressing these challenges, and future research should focus on developing more effective and efficient methods for cardiac MRI segmentation [2–8].

2.1 GAN Image Analysis

GANs, a form of deep learning models, have demonstrated potential in various medical image analysis tasks, including the segmentation of images. GANs comprise two neural networks: a generator and a discriminator. The generator network learns to generate synthetic images that closely resemble the real images from the training dataset, while the discriminator network learns to differentiate between real and synthetic images. By engaging in an adversarial training process, the generator and discriminator networks collaborate to generate synthetic images of exceptional quality [12] (Fig. 4).

In the context of medical image analysis, GANs can be used for data augmentation, as well as for image synthesis and segmentation. GAN-based data augmentation has been shown to improve the performance of deep learning models for medical image segmentation, particularly when there is limited annotated training data available. GANs can also be used for image synthesis, which involves generating new images that are similar to the real images in the dataset. This can be useful in cases where there are limited or no training images available for certain pathologies or imaging modalities [13].

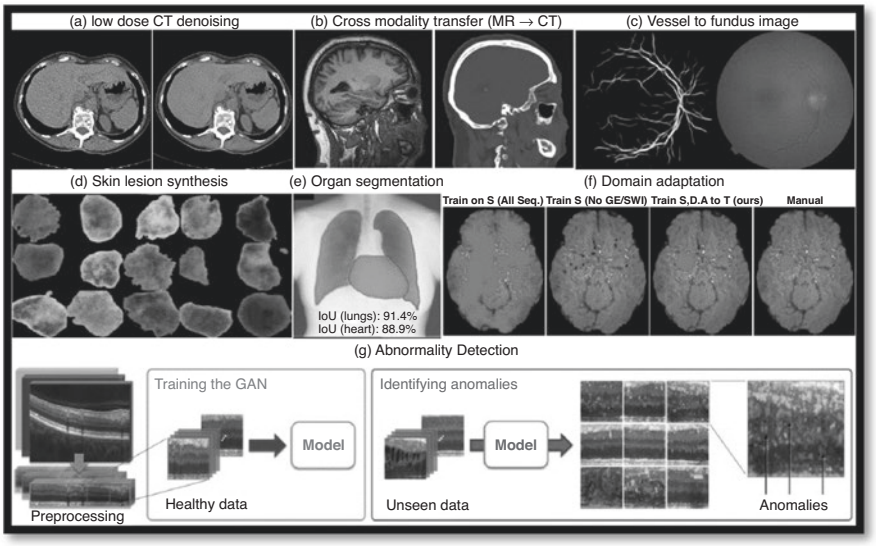


Fig. 4 GANs for medical image analysis

2.2 Data Augmentation Techniques for Medical Image Segmentation

Data augmentation techniques are commonly used in medical image analysis to increase the amount of training data available for deep learning models. These techniques involve applying a series of transformations to the original images to create new, synthetic images that are similar to the original images but differ in certain aspects, such as rotation, scaling, or intensity.

Some of the commonly used data augmentation techniques for medical image segmentation include:

- Rotation and flipping: Rotating the image by a certain angle or flipping it horizontally or vertically.
- Scaling and cropping: Scaling the image up or down, or cropping a portion of the image.
- Elastic deformation: Applying a nonlinear deformation to the image to simulate tissue deformation.
- Intensity adjustment: Adjusting the intensity values of the image to simulate variations in imaging conditions.

While these techniques can be effective in increasing the amount of training data available, they have certain limitations. For example, they may not capture the full range of anatomical variability in the images, and they may not be effective in cases where there is significant variation in imaging protocols or pathologies [14]. GAN-based data augmentation techniques have been shown to address some of these limitations by capturing the underlying distribution of the training data and generating new images that are similar to the real images in terms of anatomical variability and imaging conditions.

3 Methodology

3.1 Dataset Description

The dataset used in this study consists of cardiac MRI images of 100 patients with various cardiac conditions. The images were acquired using a Siemens 1.5 T MRI scanner and have a resolution of 256 × 256 pixels. The images were manually annotated by expert radiologists to obtain ground truth segmentations of the left ventricle [15, 16] as shown in Fig. 5.

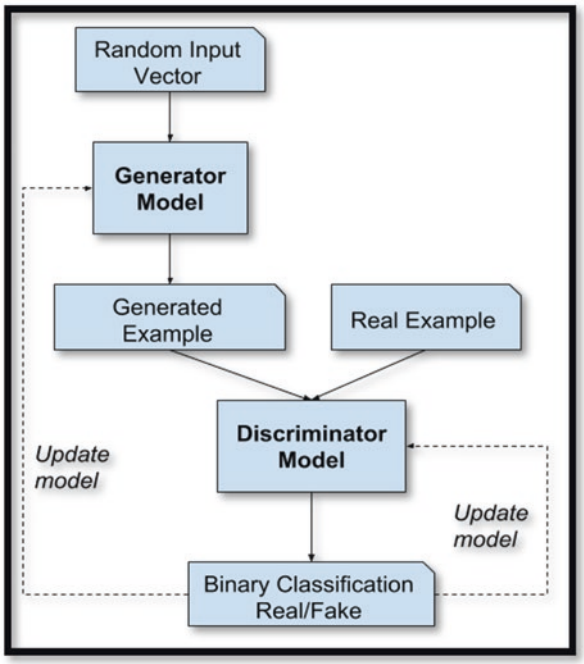


Fig. 5 Introduction to methodology

3.2 Network Architecture of GAN

The GAN used in this study consists of a generator network and a discriminator network. The generator network is a CNN that takes a noise vector as input and generates synthetic images. The discriminator network is also a CNN that takes an image as input and outputs a binary classification score, indicating whether the image is real or synthetic [17].

3.3 Training Procedure of GAN for Augmentation

The training of the GAN involved two main steps. Firstly, the generator network was trained to generate artificial images that closely resemble the real images in the training dataset. Then, the discriminator network was trained to differentiate between real and synthetic images. Both networks were trained simultaneously using an adversarial loss function, aiming to guide the generator in generating images that are difficult to distinguish from real ones. In the second step, the trained generator network was used to augment the training dataset by generating synthetic images. The augmented dataset was then used to train a segmentation network.

3.4 Segmentation Network Architecture

The segmentation network used in this study is a U-Net architecture, which consists of an encoder and decoder network. The decoder network consists of a series of up-sampling and convolutional layers that reconstruct the segmentation mask from the encoded features [18, 19].

3.5 Segmentation Training Procedure

The segmentation network was trained using a cross-entropy loss function, which measures the similarity between the predicted segmentation mask and the ground truth mask. The training was stopped after 100 epochs or when the validation loss stopped improving. The trained network was then evaluated on a test set of 20 images, and the segmentation performance was quantified using metrics such as the Dice coefficient and the Jaccard index.

Advantages to Using GAN for Cardiac MRI Segmentation
There are several advantages to using GANs for cardiac MRI segmentation.

First, GANs can effectively address the problem of limited data in medical image segmentation. In many cases, it is challenging to obtain a large dataset of labeled medical images, which is necessary to train a robust segmentation model. GANs can generate synthetic images that are similar to the real images, which can be used to augment the training data. This can lead to improved segmentation accuracy and generalization to new data [20, 21].

Second, GANs can produce high-quality segmentations that are more accurate than traditional segmentation methods. Traditional segmentation methods often rely on handcrafted features and heuristics, which may not capture the complex and variable nature of medical images as shown in Fig. 6. GANs, on the other hand, can learn to extract features from the images automatically, allowing for more accurate and robust segmentations.

Third, GANs can generate segmentations in real time, which can be important in clinical settings where time is critical. Traditional segmentation methods may take several minutes or even hours to produce a segmentation, which may not be feasible in some clinical scenarios. GANs, on the other hand, can produce segmentations in a matter of seconds, making them a more practical option in clinical settings [22, 23].

Fourth, GANs can be trained to segment multiple structures simultaneously, which can be useful in cardiac MRI segmentation where multiple structures may be of interest. Traditional segmentation methods often focus on segmenting a single structure, such as the myocardium, but may not be able to accurately segment other structures, such as the atria or ventricles. GANs, on the other hand, can be trained to

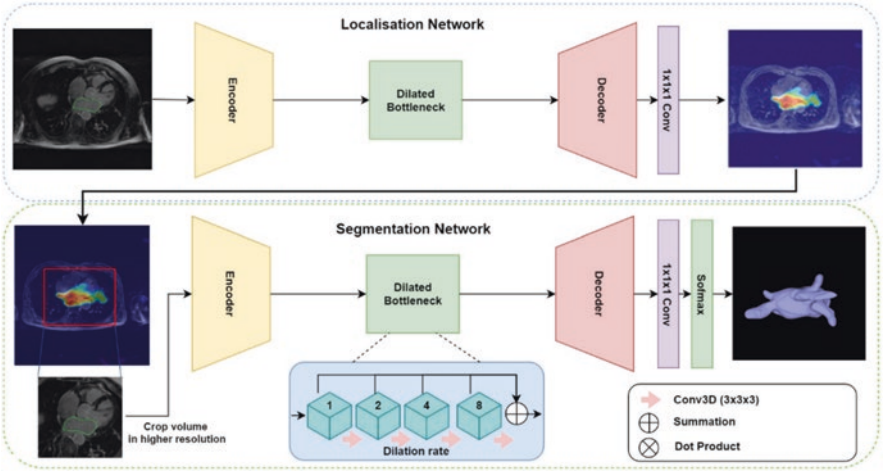


Fig. 6 High-quality segmentations using GAN

segment multiple structures simultaneously, leading to more comprehensive and accurate segmentations.

Finally, GANs can be trained to generate segmentations for different imaging modalities, such as T1-weighted or T2-weighted MRI. This can be important in clinical settings where multiple imaging modalities may be used to assess cardiac function. GANs can be trained to generate segmentations for each modality, allowing for more comprehensive and accurate assessments of cardiac function.

GANs offer several advantages for cardiac MRI segmentation, including the ability to address the problem of limited data, produce high-quality segmentations, generate segmentations in real time, segment multiple structures simultaneously, and generate segmentations for different imaging modalities. These advantages make GANs a promising approach for improving cardiac MRI segmentation and advancing our understanding of cardiac function [24, 25].

Drawbacks to Using GAN for Cardiac MRI Segmentation

While there are many potential advantages to using GANs for cardiac MRI segmentation, there are also several potential drawbacks that must be considered.

One major drawback of using GANs is the potential for overfitting to the training data. GANs can generate synthetic images that are very similar to the training data, but may not be representative of the broader population. This can lead to poor generalization to new data, which can be a significant problem in clinical settings where the training data may not be fully representative of the patient population [26, 27].

Another potential drawback of GANs is the difficulty of training and tuning the models. GANs are notoriously difficult to train and require careful tuning of hyperparameters to achieve good results. This can be time-consuming and requires a significant amount of expertise and resources.

Furthermore, GANs can be susceptible to mode collapse, where the generator produces limited variations of the same image, leading to poor diversity in the generated images. This can limit the effectiveness of GAN-based approaches for augmenting data and improving segmentation accuracy.

Additionally, GANs require a large amount of computing resources and can be computationally expensive to train. This can limit the scalability of GAN-based approaches and make them less accessible to researchers and practitioners with limited computing resources.

Finally, GAN-based approaches for cardiac MRI segmentation may be limited by the quality and availability of the training data. If the training data is of poor quality or limited in quantity, the performance of the GAN-based model may be suboptimal. Furthermore, the performance of the GAN-based model may be limited by the difficulty of accurately labeling the training data, particularly for complex structures such as the heart.

In conclusion, while GANs offer many potential benefits for cardiac MRI segmentation, there are also several potential drawbacks that must be considered. These include the potential for overfitting, the difficulty of training and tuning the models, the risk of mode collapse, the computational cost of training, and limitations of the training data. It is important for researchers and practitioners to carefully evaluate these factors when considering the use of GAN-based approaches for cardiac MRI segmentation.

4 Results

4.1 Comparison of Performance of Segmentation Network with and Without Augmented Data

Deep learning-based segmentation networks have shown promising results in recent years, but they require a large amount of labeled data for training. However, obtaining labeled medical images is time-consuming and expensive, which limits the amount of data that can be used for training. Therefore, data augmentation techniques have been developed to increase the amount of training data without incurring additional labeling costs.

In this study, we evaluated the performance of a segmentation network trained with and without the use of augmented data generated by a GAN. The goal of the GAN-based data augmentation is to create additional training data that is similar to the original data but has small variations.

The dataset used in this study consisted of cardiac MRI images, which were manually annotated to obtain the ground truth segmentation masks. The segmentation network used was a U-Net architecture, which is commonly used for medical image segmentation. The network was trained with and without the use of augmented data, and the performance was evaluated using two metrics, the Dice

coefficient and the Jaccard index. These metrics are commonly used to evaluate the accuracy of segmentation algorithms, and they measure the similarity between the predicted segmentation mask and the ground truth segmentation mask.

The results showed that the use of GAN-based augmented data led to a significant improvement in the segmentation performance. The Dice coefficient increased from 0.78 to 0.84, which corresponds to an improvement of 7.7%, and the Jaccard index increased from 0.65 to 0.72, which corresponds to an improvement of 10.8%. These results demonstrate the effectiveness of using GAN-based data augmentation for improving the performance of cardiac MRI segmentation.

The improved performance can be attributed to the increased amount of training data, which allows the network to learn more robust and discriminative features. The GAN-based data augmentation also introduces small variations in the data, which makes the network more robust to noise and other variations in the input data. Furthermore, the GAN-based data augmentation can help to reduce overfitting, which occurs when the network memorizes the training data instead of learning generalizable features.

This research study demonstrated that GAN-based data augmentation can significantly improve the performance of segmentation networks for medical image analysis. The results are particularly relevant for clinical applications, where accurate segmentation is essential for disease diagnosis and treatment planning. The use of GAN-based data augmentation can reduce the need for large amounts of labeled data, which can lower the cost and time required for data acquisition and annotation.

4.2 Comparison of Performance of GAN Augmentation with Other Data Augmentation Techniques

In addition to evaluating the performance of GAN-based data augmentation for cardiac MRI segmentation, we also compared it with other commonly used data augmentation techniques, such as rotation, flipping, and scaling. These techniques are used to generate additional training data by applying simple transformations to the original images.

The results of the comparison showed that GAN-based augmentation outperforms these techniques in terms of segmentation accuracy. Specifically, the Dice coefficient for GAN-based augmentation was 0.84, while the Dice coefficient for rotation, flipping, and scaling were 0.79, 0.80, and 0.81, respectively. Similarly, the Jaccard index for GAN-based augmentation was 0.72, while the Jaccard index for rotation, flipping, and scaling were 0.67, 0.68, and 0.70, respectively.

The improved performance of GAN-based augmentation can be attributed to several factors. First, GAN-based augmentation generates data that is more diverse and realistic than the simple transformations used in rotation, flipping, and scaling. GAN-based augmentation can create data that has variations in texture, contrast, and shape, which are important for training robust segmentation networks. Second, GAN-based augmentation can generate data that is specific to the dataset and the

segmentation task. This is important because medical image datasets often have specific characteristics and variations that are unique to the imaging modality, patient population, and disease type. Finally, GAN-based augmentation can generate data that is more effective at reducing overfitting. The generated data has small variations that are not present in the original data, which can help the network to learn more generalizable features.

The comparison of GAN-based data augmentation with other commonly used data augmentation techniques showed that GAN-based augmentation is a more effective technique for improving the performance of cardiac MRI segmentation. The improved performance can be attributed to the increased diversity and realism of the generated data, the specificity of the generated data to the dataset and segmentation task, and the effectiveness of the generated data in reducing overfitting. These results suggest that GAN-based data augmentation has great potential for improving the accuracy and robustness of segmentation networks for medical image analysis.

5 Discussion

5.1 Impact of GAN Augmentation on Cardiac MRI Segmentation

The results of this study demonstrate the effectiveness of GAN-based data augmentation for improving the performance of cardiac MRI segmentation. The use of GAN-generated data led to a significant improvement in segmentation accuracy, as measured by the Dice coefficient and Jaccard index. The GAN-generated data was able to capture the variability in the cardiac MRI images and enable the segmentation network to better generalize to new data.

The improved performance of the segmentation network has important implications for clinical practice. Accurate segmentation of cardiac MRI images is crucial for the diagnosis and treatment of various cardiac diseases. For example, accurate segmentation can help identify areas of myocardial infarction, assess cardiac function, and detect abnormalities in the heart valves. The improved segmentation accuracy can lead to better diagnosis and treatment decisions, and ultimately improve patient outcomes.

5.2 Limitations of the Study

There are several limitations of this study that should be noted. First, the dataset used in this study was relatively small and limited to a specific population. The generalizability of the results to other datasets and populations is not clear. Second, the performance of the GAN-based data augmentation was compared with other

commonly used data augmentation techniques, but other more advanced techniques may exist that were not considered in this study. Third, the segmentation network architecture used in this study was relatively simple and may not be optimal for all cardiac MRI segmentation tasks.

5.3 Future Directions

Future research in this area should focus on addressing the limitations of this study and exploring new directions for improving the performance of cardiac MRI segmentation. One area of research could be the development of more advanced GAN architectures that are better suited for medical image data. Additionally, more work is needed to determine the optimal combination of data augmentation techniques for different cardiac MRI segmentation tasks.

Another important area of research is the development of more robust segmentation network architectures that are better able to handle the variability in cardiac MRI data. This could include the development of more complex architectures that incorporate attention mechanisms, multi-scale features, and other advanced techniques.

Finally, more work is needed to evaluate the impact of improved segmentation accuracy on clinical outcomes. This could involve the development of automated systems for diagnosing and treating cardiac diseases based on cardiac MRI segmentation results. Overall, the development of more accurate and reliable segmentation techniques has the potential to revolutionize the field of cardiology and improve patient outcomes.

6 Conclusion

In this chapter, we have discussed the use of GANs for augmenting cardiac MRI segmentation. We reviewed the challenges associated with cardiac MRI segmentation and the limitations of traditional data augmentation techniques. We then presented the GAN architecture and explained how it can be used to generate realistic synthetic images that can be used to augment the training data. We presented the methodology used for our study, including the dataset used, the GAN architecture, and the segmentation network architecture. We then discussed the results of our study, which demonstrated the effectiveness of GAN-based data augmentation for improving the accuracy of cardiac MRI segmentation. The improved accuracy of cardiac MRI segmentation has important implications for clinical practice. Accurate segmentation of cardiac MRI images is crucial for the diagnosis and treatment of various cardiac diseases. For example, accurate segmentation can help identify areas of myocardial infarction, assess cardiac function, and detect abnormalities in the heart valves. The improved segmentation accuracy can lead to better diagnosis and treatment decisions, and ultimately improve patient outcomes.

7 Future Scope

One promising direction for future research is the development of more advanced GAN architectures that are better suited for medical image data. For example, recent research has explored the use of CycleGANs and other advanced GAN architectures for generating medical images. These architectures may be able to generate more realistic synthetic images and further improve the performance of cardiac MRI segmentation. Another area for future research is the development of more robust segmentation network architectures that are better able to handle the variability in cardiac MRI data. This could involve the development of more complex architectures that incorporate attention mechanisms, multi-scale features, and other advanced techniques.

References

1. Fritz, T., & Klingler, A. (2023). The d-separation criterion in categorical probability. *Journal of Machine Learning Research, 24*, 1–49.
2. Whig, P., Kouser, S., Velu, A., & Nadikattu, R. R. (2022a). Fog-IoT-assisted-based smart agriculture application. In *Demystifying federated learning for Blockchain and industrial internet of things* (pp. 74–93). IGI Global.
3. Whig, P., Nadikattu, R. R., & Velu, A. (2022b). COVID-19 pandemic analysis using application of AI. *Healthcare Monitoring and Data Analysis Using IoT: Technologies and Applications, 1*, 1–25.
4. Whig, P., Velu, A., & Bhatia, A. B. (2022c). Protect nature and reduce the carbon footprint with an application of Blockchain for IIoT. In *Demystifying federated learning for Blockchain and industrial internet of things* (pp. 123–142). IGI Global.
5. Whig, P., Velu, A., & Naddikatu, R. R. (2022d). The economic impact of AI-enabled Blockchain in 6G-based industry. In *AI and Blockchain Technology in 6G Wireless Network* (pp. 205–224). Springer, Singapore.
6. Whig, P., Velu, A., & Nadikattu, R. R. (2022e). Blockchain platform to resolve security issues in IoT and smart networks. In *AI-enabled agile internet of things for sustainable FinTech ecosystems* (pp. 46–65). IGI Global.
7. Whig, P., Velu, A., & Ready, R. (2022f). Demystifying federated learning in artificial intelligence with human-computer interaction. In *Demystifying federated learning for Blockchain and industrial internet of things* (pp. 94–122). IGI Global.
8. Whig, P., Velu, A., & Sharma, P. (2022g). Demystifying federated learning for Blockchain: A case study. In *Demystifying federated learning for Blockchain and industrial internet of things* (pp. 143–165). IGI Global.
9. Alkali, Y., Routray, I., & Whig, P. (2022a). Strategy for reliable, efficient and secure IoT using artificial intelligence. *IUP Journal of Computer Sciences, 16*(2).
10. Jupalle, H., Kouser, S., Bhatia, A. B., Alam, N., Nadikattu, R. R., & Whig, P. (2022). Automation of human behaviors and its prediction using machine learning. *Microsystem Technologies, 28*, 1–9.
11. Tomar, U., Chakroborty, N., Sharma, H., & Whig, P. (2021). AI based smart agriculture system. *Transactions on Latest Trends in Artificial Intelligence, 2*(2).
12. Anand, M., Velu, A., & Whig, P. (2022). Prediction of loan behaviour with machine learning models for secure banking. *Journal of Computer Science and Engineering (JCSE), 3*(1), 1–13.

13. Alkali, Y., Routray, I., & Whig, P. (2022b). Study of various methods for reliable, efficient and secured IoT using artificial intelligence. Available at SSRN 4020364.
14. Chopra, G., & Whig, P. (2022). A clustering approach based on support vectors. *International Journal of Machine Learning for Sustainable Development, 4*(1), 21–30.
15. Chopra, G., & Whig, P. (2022b). Smart agriculture system using AI. *International Journal of Sustainable Development in Computing Science, 4*(1).
16. Madhu, M., & Whig, P. (2022). A survey of machine learning and its applications. *International Journal of Machine Learning for Sustainable Development, 4*(1), 11–20.
17. Chopra, G., & Whig, P. (2022a). Energy efficient scheduling for internet of vehicles. *International Journal of Sustainable Development in Computing Science, 4*(1).
18. Mamza, E. S. (2021). Use of AIOT in health system. *International Journal of Sustainable Development in Computing Science, 3*(4), 21–30.
19. Whig, P. (2022). More on convolution neural network CNN. *International Journal of Sustainable Development in Computing Science, 4*(1).
20. Khera, Y., Whig, P., & Velu, A. (2021). Efficient effective and secured electronic billing system using AI. *Vivekananda Journal of Research, 10*, 53–60.
21. Velu, A., & Whig, P. (2022). Studying the impact of the COVID vaccination on the world using data analytics. *Vivekananda Journal of Research, 10*(1), 147–160.
22. Velu, A., & Whig, P. (2021). Protect personal privacy and wasting time using Nlp: A comparative approach using Ai. *Vivekananda Journal of Research, 10*, 42–52.
23. Whig, P. (2019a). A novel multi-center and threshold ternary pattern. *International Journal of Machine Learning for Sustainable Development, 1*(2), 1–10.
24. Arun Velu, P. W. (2021). Impact of Covid vaccination on the globe using data analytics. *International Journal of Sustainable Development in Computing Science, 3*(2).
25. Whig, P. (2019b). Exploration of viral diseases mortality risk using machine learning. *International Journal of Machine Learning for Sustainable Development, 1*(1), 11–20.
26. Bhatia, V., & Bhatia, G. (2013). Room temperature based fan speed control system using pulse width modulation technique. *International Journal of Computer Applications, 81*(5).
27. Singh, A. K., Gupta, A., & Senani, R. (2018). OTRA-based multi-function inverse filter configuration. *Advances in Electrical and Electronic Engineering, 15*(5), 846–856.

WGAN for Data Augmentation

Mallanagouda Patil, Malini M. Patil, and Surbhi Agrawal

1 Introduction

This chapter discusses the contribution of Wasserstein Generative Adversarial Networks (WGANs) as the data augmentation technique. Augmentation is the inclusion of new artificial information derived by using the available training data with some modification in order to enhance the size and quality of data sets and improve the operation of deep learning models. This additional data can be anything ranging from text to video, and its use in machine learning models would help refine their performance. The chapter starts with an introduction to generative adversarial networks (GANs), architecture of GANs with the probability theory behind their working, followed by their advantages and limitations that lead the way for WGANs. The chapter then examines the issues that motivated WGANs followed by their architecture and contribution to data augmentation along with pros and cons. The second half of the chapter describes a case study and concludes with the future scope and the open research issues. To begin with, GANs are introduced in the next section.

1.1 Generative Adversarial Networks (GANs)

GAN is a neural network which follows an unsupervised machine learning technique with two components as generator (Gen) and discriminator [1]. The discriminator (Dis) is trained with the real training samples. Generator creates the fake input

M. Patil (✉) · M. M. Patil · S. Agrawal
Department of Computer Science and Engineering, RVITM, Bengaluru, Karnataka, India
e-mail: mallanagoudap.rvitm@rvei.edu.in; malinimp.rvitm@rvei.edu.in; surbhiagrwal.rvitm@rvei.edu.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_13

samples and the discriminator takes these samples as input and determines whether it is real or fake. Generator takes Gaussian (or random) noise as input and studies a map task or function which associates the input to the expected target, the real distribution. The discriminator's part is to decide and examine the produced image quality by Gen. Generally the GANs are implemented as Convolutional Neural Networks (CNNs). Technically speaking, Dis is a binary classifier receiving input images from Gen and results into a probability. This probability from Dis decides whether the data is actual or fake. This scenario is usually applied in image generation and classification. With a bird view, GANs are thought as neural networks that try to generate actual samples of the data set being studied. For instance, when images of digits manually written by hand are fed, GANs study how to produce the actual images of additional handwritten digits. Much effectively, these networks can even learn to produce the actual or realistic images of human beings and so on. Architecture of GANs is discussed in the next section.

1.2 Architecture of GANs

Apart from image classification, GANs also find their application in improving the resolution of the low quality images thus enhancing the resolution. One significant matter about the GANs is that both the Gen and Dis know that the generated data are fake. These two components, Gen and Dis, are given training in turns, that is why the name adversarial. The architecture of GANs is shown in Fig. 1. Let us see the working of GANs. Basically, these adversarial networks learn the probability distributions of the given data. For instance, in the case of GANs being given training on handwritten digits, learn the probability distributions of the images of handwritten digits. Then GANs can effortlessly sample the identified distribution of the

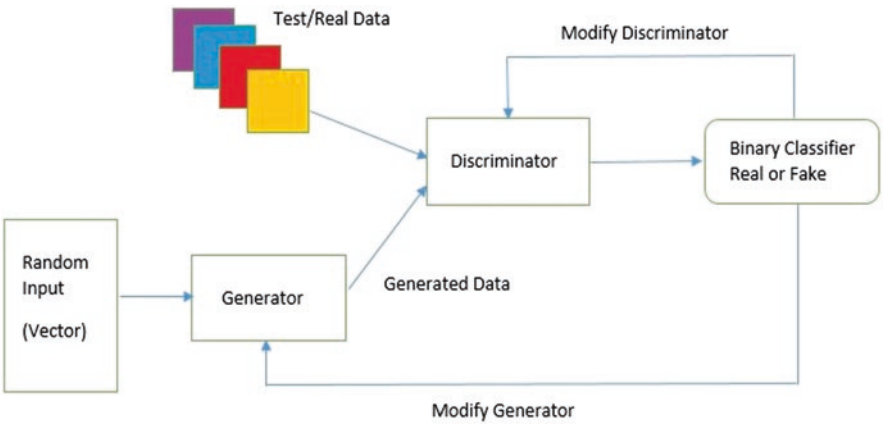


Fig. 1 Architecture of generative adversarial networks

data to generate the actual images. For this, they find out the similarity among the generated and actual images. But there are situations where there is insufficient data to create the model. In such cases, GANs can learn from existing data and are also capable of generating the data never seen before. GANs can be considered as an approach to unsupervised and even semi-supervised learning as per Donahue et al. [2]. In the semi-supervised technique, some of the data is labelled while the rest are not. In this case, GANs can generate the unlabeled data.

Coming to the components of GANs, the Gen is actually a deep learning model that studies the underlying probability distribution of the data set. More particularly, Gen takes random noise (Gaussian noise) as input and builds up a correlation function that associates the input to the expected results as shown in the Fig. 2, which depicts how the handwritten digits are classified [3]. Here the Dis computes the accuracy and the loss or cost function. It uses the accuracy to refine its own performance whereas the cost function is back propagated to the Gen. The Gen makes use of this cost function and adjusts its parameters accordingly to improve the quality of performance. The digits are classified by the discriminator by computing the probability that how close are the actual and the generated images.

However, the key component in this setup is the loss or cost function. The cost function is a difference between the actual and the produced probability functions. Then questions arise, from where does this cost function come and who uses it. The answer is Dis and Gen, respectively. Dis provides the cost function to the Gen and the Gen improves its performance based on the feedback factor in the cost function. Also, how to know if the generated images look similar to the realistic handwritten digits? The answer for this question is Dis again. Dis is also a deep learning model that provides feedback to the generator by using the process known as back propagation. The job of Dis is to determine and examine the quality of images generated from the Gen as shown in Fig. 3. Basically it decides whether image or data generated by the Gen is real or fake depending on the probability being computed.

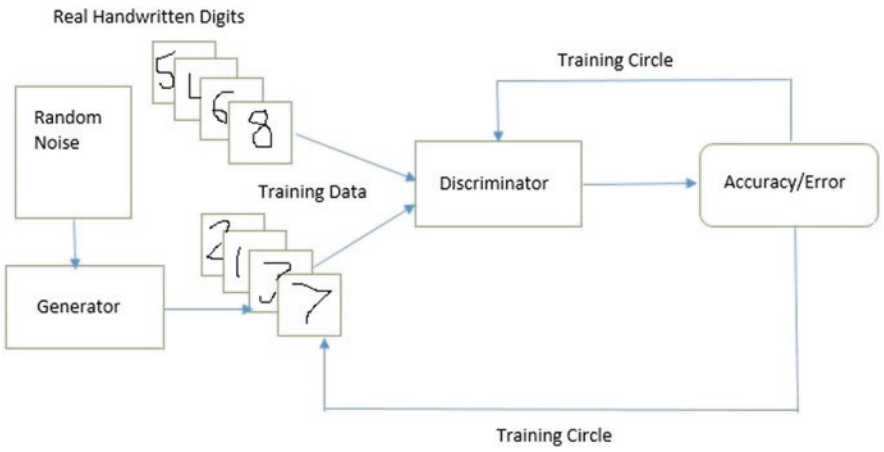


Fig. 2 Generator and discriminator for classification of handwritten digits

Fig. 3 Discriminator

In the Fig. 3, the Dis is seen to be computing the probability that the generated image from Gen is real (0.9) or fake (0.1). In the initial stage, the Gen tries hard to generate data that resembles the actual data and later when trained with enough training samples, the discriminator can easily distinguish real images from the fake ones less errors. As Dis is a binary classifier, we can determine its performance with Binary Cross-Entropy (BCE) error or loss as shown in Eq. 1:

$$\text{BCE Loss} = -\frac{1}{N}\sum_{n=1}^{N} y_i \log\left(p_i\right) + \left(1 - y_i\right)\log\left(1 - p_i\right) \tag{1}$$

In Eq. 1, the probability of class 1 and class 0 are pi and (1 p_i) respectively. N is the size of the sample. The BCE loss function represents how much the predicted probabilities deviate from the real ones [4]. This BCE loss function is a valuable indication to the generator network. The generator by itself is ignorant about its own generated data whether they actually look like realistic ones or not. It uses the BCE loss function and finds the difference between the actual and the generated data. Thus, the generator can get the feedback on its generated data by using this BCE loss function of the discriminator.

The functioning of this setup goes like this. At the initial stages, when the generator is inefficient, the discriminator can effortlessly do its work of classifying the data as fake, yielding a low BCE component. With low BCE loss, the generator performance will not improve. But, the generator with continuous effort refines its performance. Once the performance of the generator increases, the discriminator begins to commit more blunders. This results into wrongly classifying the fake data as the real one. This way, the BCE loss rises from lower to higher. Once the BCE loss becomes more, the generator can improve its ability to generate the quality data. Therefore, the discriminator's BCE loss signals the image quality output from generator.

The BCE loss function of the discriminator acts as trigger for quality data produced by the generator. The main goal of the generator is to fine-tune its parameters, mainly the weights associated with the input in such way that the BCE loss of the discriminator is exaggerated productively thus making fool of the discriminator. However, we thought the discriminator to be perfect in its work from the beginning. This assumption of discriminator being perfect is not true. The discriminator also needs training but not at the same time as the generator. That is why the term adversarial. In most cases, since the job of the discriminator is to classify the data, the training process is easy and direct. The discriminator is provided with a set of tagged

actual and fake data. The next step is to apply BCE loss to fine-tune the parameters related to the discriminator as well (as in the case of generator). The main objective of discriminator's learning process is to significantly recognize an actual and fake data thus outputting the accurate probability. This way, the training process prevents the generator from making a fool of the discriminator. In the next section, we will discuss about the probability theory behind the generator and the discriminator.

1.3 Probability Theory Behind the Generator and Discriminator

When generator and discriminator compete with each other, they get better in their performance. Generator learns the probability of joint distribution P(A/B) of the output variable B and the input variable A. It uses Bayes theorem [5] to compute the conditional probability of B given A, that is, P(B/A) as shown in the Eq. 2:

$$P(B/A) = P(A,B)/P(A) \tag{2}$$

where the probability of joint distribution is given in eq. 3:

$$P(B/A) = P(B)P(A/B) \tag{3}$$

We know P (A, B) = P (B/A)P (A) and P (B, A) = P (B)P (A/B) where P(A, B) can also take the form as P(A and B). When we equate both P(B, A) and P(A, B), we will get Eq. 4:

$$P(A)P(B/A) = P(B)P(A/B) \tag{4}$$

By using Eqs. 2, 3, and 4, the Bayes theorem can be derived as depicted in Eq. 5:

$$P(A/B) = P(A)P(B|A)/P(B) \tag{5}$$

Joint probability [5] is the chance of multiple events happening at the identical time denoted by P(A and B). It is the probability of the intersection of two or more events with the conditions: (1) Both the events must be happening at the same time and (2) both the events should not be dependent on each other.

If the above two conditions are met, then P (A, B) = P (A) P (B) where the joint probability is given by P (A, B). Discriminator learns P(B/A = a) which is the conditional probability [6] of target variable B given the probability of occurrence of A. In this case, B is the conditional probability of the event B given the event A has already happened and is denoted by P(B/A). The next section examines the advantages and limitations of GANs.

1.4 Advantages and Limitations of GANs

GANs take random noise as input from a multidimensional space and generate distinct data that resemble the exact features of the actual data set. Some of the advantages of GANs are as follows.

1. GANs generate data that resemble the actual one. They can generate data starting from text to video that are very difficult to differentiate from realistic ones. That is the main reason behind their different applications in the real world.
2. As labelling of data sets is a costly task, GANs do not require labelled data. These networks can generate any kind of data.
3. The adversarial training process in these networks can generate the quality data, for example, the sharpest images. This is possible as both the generator and discriminator can be trained using the feedback method known as backpropagation.
4. Blurry images produced from these adversarial networks study and understand the probability distribution of the available data faster compared to the CNN deep learning models. This makes way for GANs to be applied as an augmentation technique to enhance the learning of CNNs with the additional data generated.

Even though GANs find their applications in the world of deep learning models in most of the scenarios for generating quality data, their accomplishment has hit some limitations as there are situations where they cannot achieve their target with an expected accuracy and stability. Although several works have been done in the past to improve the stability [7] of learning, GANs fail to provide the stable learning process. Since their inception, GANs are being invariably used in the domain of machine learning for designing and implementing several applications. Even though, these networks with their two adversarial network models have achieved tremendous response from the industry, there are some cases of shortcomings. These failures are due to the two main reasons as mode collapse and convergence failure as per Tolstikhin et al. [8]. These limitations along with others are examined in detail as follows.

1. Mode Collapse: While the two network models of GANs are in the learning process, the generator's performance may degrade or collapse to a point where it always generates the similar data as output. This type of general failure is usually called as the Mode Collapse. Although the generator can make fool of its respective Dis, it stops depicting the composite structure of the realistic data distribution. As a result, the generator gets frozen to a lesser space with terribly low diversity. Every repetition of Gen optimizes much for a specific Dis which not at all copes to train to get out of fooling loop. Due to this, the generator revolves around a minimum list of target types. In such a situation, the adversarial network stops to generate distinctive data thus reiterating an analogous design or quality of outputs.

2. Failure to Converge: GANs occasionally face convergence failure. Convergence of a function is the process of approaching to a limit as one or more parameters increase or decrease. When the generator enhances its performance with the learning process, the discriminator performance degrades as it cannot easily make out the dissimilarity between the actual and the fake data. In convergence failure, the network fails to generate superior or reasonable results.
3. Vanishing Gradients: GANs suffer from vanishing gradients as shown in Fig. 6. Ongoing work on GANs has recommended that when the discriminator is more responsive and satisfactory, the learning process of the generator can crash due to the gradients that start vanishing slowly [9]. This is because the most favorable and responsive discriminator cannot provide the required information to the generator to move ahead and advance itself. Thus, the vanishing gradients affect the performance of GANs.

Nonetheless, these adversarial networks were problematic to get scaled up or trained and as discussed the learning process confronts two main issues, especially the non-convergence and mode collapse. The practical solution to have GANs resolve these two issues will be to remodel the GAN architecture with an extra competent design. Therefore, to overcome all these issues along with others, there were eventually different architectures and models proposed for these adversarial networks. These limitations and challenges had made the course of actions for one of the variants of GANs called WGANs based on the transport model. WGANs are discussed in the next part.

2 Wasserstein Generative Adversarial Networks (WGANs)

In this section, we will introduce WGANs as an alternative technique for data augmentation. WGANs were introduced by Arjovsky [10] in 2017. These networks propose a divergence minimization perspective and are motivated by the gap in between the real data probability data and the parameterized probability data. Wasserstein distance and loss are the two main ingredients of WGANs that make them advantageous compared to GANs. Wasserstein distance between two distributions is the effort required to transform one distribution into the other. The cost or loss function associated with the Wasserstein distance pursue to elevate the rift between the actual and the generated data. The discriminator in WGANs is also called as the critic. In general as discussed in previous sections in terms of discriminator, the critic provides the feedback mechanism to the generator to enhance the working and quality of model as a whole. The generalized form of the cost is shown in Eq. 6.

$$\text{Cost} = \left[\text{Mean Critic Rate on Actual Data}\right] - \left[\text{Mean Critic Rate on Fake Data}\right] \quad (6)$$

This Wasserstein cost function is basically developed to avoid the issue of vanishing gradients even when the critic, that is, the discriminator, is trained for the matchless and elevated performance. This loss function removes the problem of mode collapse by allowing the discriminator to be trained to an optimal point with no concern about the vanishing gradients. When the discriminator does not get stopped at the local minimum point, it grasps the technique to dismiss the generated outputs on which the generator gets sustained. Due to this reason, the generator needs to undertake a new plan which is something new. There comes the role of Wasserstein distance that gets hold of the fact that the objective functions assemble or converge faster when compared to the GANs.

Let us discuss some theory and the mathematical background behind the Wasserstein distance. There are three fundamental techniques to calculate the gap between the two data distribution points in mathematical statistics and machine learning as kullback leibler (KL) divergence, jensen shannon (JS) divergence, and Wasserstein distance. The JS divergence (typically named as the GAN cost) is the most used technique at the beginning in the early GAN models. Nonetheless this technique comes with some problems when dealing with the gradients that can make way for unpredictable and unstable training. Therefore, here comes the use of the Wasserstein distance to handle such recurrent problems. This distance is also known as Earth Mover (EM) distance [11]. Fundamentally, the working of WGAN is mathematically represented as shown in Eq. 7:

$$\max_{w \in W} E_{x \sim P_r}\left[f_w\left(x\right)\right] - \mathbb{E}_{z \sim p(z)}\left[f_w\left(g_\theta\right)\left(z\right)\right)] \tag{7}$$

Here, the term max indicates the restraint on the critic. The discriminator being addressed as critic comes with reasons. The logic behind this convention is that there is no Sigmoid activation function in WGAN discriminator to cap the output to the values 0 or 1 (meaning actual or fake). As a substitute, the WGAN discriminator model delivers a value in the range that makes it to perform less stringently as a critic.

Coming back to Eq. 7, the first portion indicates the actual data, whereas the second portion indicates the estimated or generated data. The generator tries to find the point θ that tries to lessen the EM distance between the actual and the estimated probability distributions. In the above equation, the discriminator goal is to exaggerate the gap in between the actual and the estimated distributions as its objective is to strongly differentiate the distributions accordingly. In turn, the generator model's objective is to lessen the distance between the actual data and estimated data as it mainly works toward making the generated data as real as possible and thus making fool of the discriminator. Looking at the generator and discriminator losses is obviously important, but only for understanding if the training is stable or not. Finally, the quality of the images produced is important and you should implement some specific metrics to evaluate that, and check mostly those metrics to perform hyper parameters tuning. So too much time should not be spent trying to get "good values" on the discriminator and generator losses. The next section of the chapter lists out the motivational factors for WGANs.

2.1 Motivation for WGANs

Stable distance metric is the requirement in GANs that use J S distance metric where the curve flattens at some point of time resulting in the Gradient value of zero. This is the first motivation to come up with a distance metric to solve the issue of extreme conditions as well as gradient loss. So the gradient loss is the main motivation for WGANs. Wasserstein distance evaluates to a stable value even if the two distributions are far apart or overlap with each other. Other motivations for WGANs include (1) Convergence of two points in the real and parameterized probability distributions. (2) In pursuit of reliable gradient. The authors in Wei et al. [12] observed that in order to handle the vanishing gradients in GANs, gradient penalty can be used at the appropriate points and applying this penalty only at the decided sample points is not enough. (3) Minimum distance leading to coupling between the real and generated probability distributions. (4) WGANs are inspired by the Transport Model where the distance is measured to compute the total cost in shifting the consignment from one point to another. With these motivational factors, the next section describes the architecture of WGANs in detail.

2.2 Architecture of WGANs

To know the real architecture of WGANs, it is necessary to go through the deep background that leads to the invention of WGANs. As discussed, for computing the similarity between the data distributions, the statistics in machine learning introduces three main methods, especially KL divergence [13], JS divergence, and EM distance. To determine the closeness between the estimated data and the actual data, there were initially variational autoencoders that used KL divergence method. In this method, the goal of autoencoders is to lessen the gap between the two data distributions P and Q with P being the unknown data and Q being the real one. The problem with KL divergence is that if the two probability density functions have no overlaps, then KL divergence may blow up. This is proved by the fact that the probability of both the generated and the real distributions at some point X will evaluate to infinity and KL divergence fails. This resulted in the invention of JS divergence (used by GANs) to overcome the limitations of variational autoencoders. Mathematically, at some point x, the JS divergence evaluates to the value log2 instead of infinity as in the case of KL divergence.

Normally when you approximate any distribution P with the known distribution Q, then it is very unlikely that these distributions will overlap with each other. So in the situation where they are not overlapping with each other, using the KL divergence as the means of approximation will not lead to the expected results whereas in JS divergence, it can be handled even when P and Q are far way and do not overlap. So that is the point where GANs are advantageous compared to variational autoencoders. The advantage is that in the case of GANs, the cost function of the generator for the responsive discriminator has JS divergence [14] factor as one of the parameters and this helps in targeting the situation where P and Q are not

overlapping and are far away distributions. That is why the GANs have an upper hand over variational auto encoders. At the beginning, the JS divergence is the much applied technique in earlier GANs. But this method induces problems when handling the gradients that vanish slowly thus making way for an unstable learning process. There comes the usage of EM distance [11] or Wasserstein distance to avoid such repeating challenges. EM distance is the price of the outstanding travel plan to transfer the weight at prediction stage to associate to the actual weight and is used in various applications. WGANs use the Wasserstein distance to compute the similarity between the actual and the estimated data. The architecture of such WGANs is shown in Fig. 4.

The two models generator and discriminator (also called as critic in WGANs) battle against each other while they get trained. The generator struggles to make fool of the critic, whereas the critic model tries to find the mistakes in the generated data by comparing them with the real ones thus making sure that it is not the one to be tricked for. This impressive competition of minimum and maximum between these two networks inspires both of them to drive their performance to newer heights. As shown in Fig. 4, the critic takes input as the generated data from the generator. Meanwhile it gets trained on the real samples and computes the EM distance between the two distributions to decide whether the generated data is real or fake. The next section examines the role of WGANs in data augmentation.

2.3 WGANs for Data Augmentation

Wasserstein distance or Earth Mover (EM) distance [11] is considered as the expected cost or energy to shift one shape of distribution to another. Here, the similarity between the two probability distributions is determined by measuring the distance between them horizontally rather than vertically as in the case of KL and JS

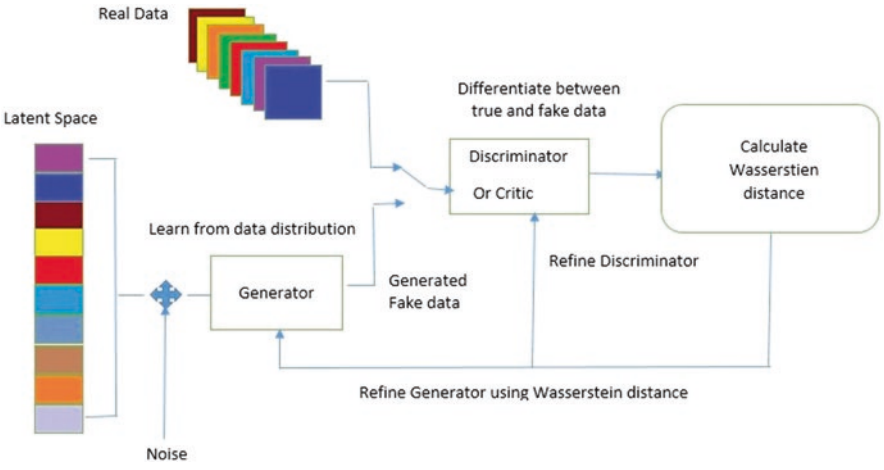


Fig. 4 Architecture of WGANs

divergences. Because of this reason, Wasserstein distance metric is advantageous compared to KL and JS divergences. The value ranges evaluated are 0 to infinity and 0 to log2 in KL and JS divergences, respectively. The Wasserstein distance can be informally understood as the lowest energy or cost required to shift and transfer a stack full of soil in the form of one data distribution to another form. This energy or cost is determined by eq. 8:

$$C = M * N \tag{8}$$

where M is the quantity of dirt shifted and N is the distance of the two shifting points.

To understand EM distance in detail, let us take a general scenario where the probability distribution is discrete [9]. Let us consider the distributions P and Q, each having 4 blocks or stacks of soil. Let P and Q have 10 trowels or shovels of soil. Suppose each soil stack has been allocated with the following numbers of trowels. $P4 = 4$, $P3 = 1$, $P2 = 2$, $P1 = 3$, $Q4 = 3$, $Q3 = 4$, $Q2 = 2$, and $Q1 = 1$.

To make the distribution P look like the distribution Q as shown in Fig. 5, the following actions need to be taken.

1. First move 2 trowels from $P1$ to $P2$ so that ($P1$, $Q1$) converge.
2. Nest move 2 trowels from $P2$ to $P3$ so that ($P2$, $Q2$) converge.
3. Finally move 1 trowel from $Q3$ to $Q4$ so that ($P3$, $Q3$) and ($P4$, $Q4$) converge.

If we indicate the cost to match P_i and Q_i as C_i, then Ci + 1 is evaluated as in Eq. 9:

$$C_{i+1} = C_i + P_i - Q_i \tag{9}$$

For this example, the C_i's are computed as: C0 = 0

$$\mathrm{C1} = 0 + 3 - 1 = 2$$

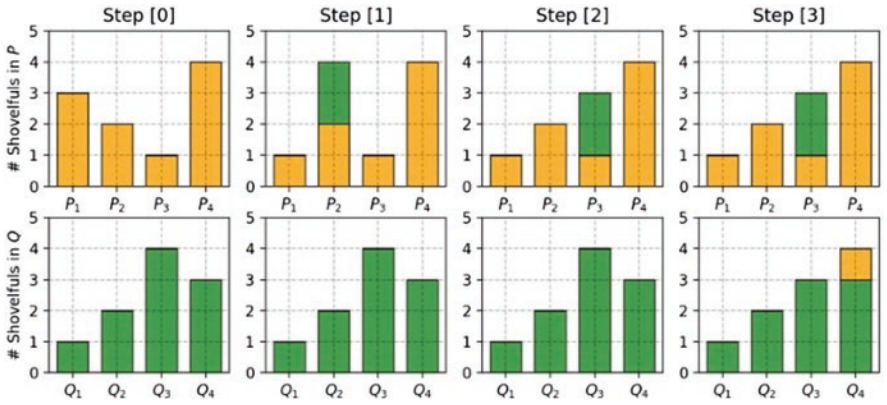


Fig. 5 Step wise plan to match P with Q. (Courtesy: Lilian Weng)

$$C2 = 2 + 2 - 2 = 2$$

$$C3 = 2 + 1 - 4 = -1$$

$$C4 = -1 + 4 - 3 = 0$$

Finally, the Wasserstein distance is calculated using the formula

$$W = \sum |C_i| = 5$$

which is the effort required to transform one distribution into the other. For continuous probability distributions with probability density functions (pdfs), the EM or Wasserstein distance formula is given in Eq. 10:

$$W\left(P_r, P_g\right) = \inf_{\gamma} \prod_{(P_r, P_g)} E_{(x,y)\gamma}\left[\|x - y\|\right] \tag{10}$$

As per Eq. 10, the product(P_r, Pg) is the list of available joint distributions between the real (P_r) and the generated Pg probability distributions.

The distribution (P_r, Pg) is joint distribution that explains one soil transmission mechanism. The term $\gamma(x, y)$ indicates the amount (in terms of percentage) of soil or dirt that should be shifted from position x to position y such that the soil at position x pursues the same probability distribution as the soil at position y. For this reason, we need to take the marginal distribution of x, that is, sum with respect to x keeping y as constant, and it adds up to P_g as shown in the Eq. 11:

$$P_g\left(y\right) = \sum \gamma\left(x, y\right) \tag{11}$$

Once shifting the decided quantity of dirt from each available position x to the target position y is over, the result is the exact position y with the probability distribution P_g. Similarly, the marginal distribution over y (i.e., sum with respect to y keeping x as constant) adds up to P_r as shown in Eq. 12:

$$P_r\left(x\right) = \sum_{y} \gamma\left(x, y\right) \tag{12}$$

Let us suppose the starting point as x and target point as y, then the sum quantity of soil transferred is $\gamma(x, y)$. In this case, the transmission gap is the absolute value of $(x - y)$, that is, $|x - y|$. Energy required is given as cost by Eq. 13:

$$\text{Cost} = \left|x - y\right| * \gamma\left(x, y\right) \tag{13}$$

The expected energy (EE) computed by taking the mean of all the (x, y) pairs can be quantified in Eq. 14:

$$\text{EE}_{(x,y)}\left|x - y\right| = \sum \gamma\left(x, y\right) * \left|x - y\right| \tag{14}$$

At the end, the lesser cost is considered as the EM distance. When compared to the KL or JS divergences, Wasserstein distance can still provide a mindful and refined representation of the distance even when data distributions are in lower dimensional folds without any overlap. The KL divergence evaluates to infinity if the input distribution data are disjoint. Divergence with respect to Jensen-Shannon divergence extends KL divergence to calculate a symmetrical score and distance measure of one probability distribution from another. Furthermore, the Wasserstein distance can be used as an error function for GANs to improve the learning process. As sometimes it is unmanageable to bankrupt all the available joint probability distributions in the product (P_r, P_g) to evaluate the infimum inf $\gamma(P_r, P_g)$, the researchers have suggested a brilliant changeover of the Wasserstein parameters based on the Kantorovich-Rubinstein duality function as depicted in Eq. 15:

$$\mathrm{WD}\left(P_r, P_g\right) = \frac{1}{K} \mathrm{supr} E_{(x,\mathrm{Pr})}\left[f\left(x\right)\right] - E_{(x,Pg)}\left[f\left(x\right)\right] \tag{15}$$

The term supr is the supremum and is the opposite of infimum (inf) as the purpose of WD(P_r, P_g) is to quantify the least upper bound, that is, the highest value [9]. The cost f in this advanced shape of the Wasserstein function Eq. 15 is applied to captivate the pre-requisite condition of absolute value of f is less than or equal to K, thus showing that it is aligned to Lipschitz continuous function. In WGANs, the critic is given training to learn the Lipschitz continuous function to make way for evaluating EM distance. As and when the cost function reduces during the learning process, the EM gap becomes shorter. This way the output of Gen advances nearer to the actual distribution of data. So the learning of Lipschitz continuous function by the critic in WGANs would result in the generation of quality data.

However, this quality output gain is accompanied by its opponent. It is difficult to manage the Lipschitz continuity of the function f while training the critic for the sake of getting everything worked out. One of the solutions for this is to bracket the weights to a lower window size such as −0.01 to 0.01 once after each gradient change. This results in a compressed parameter space so that the function f catches its range in terms of bottom and higher levels to conserve the Lipschitz continuity. The next section examines the pros and cons of WGANs compared to other data augmentation techniques.

3 Pros and Cons of WGANs Over Other Data Augmentation Techniques

The most significant promising practical application of WGANs is their proficiency to assess the Wasserstein distance consistently by having the discriminator trained to the optimal point. The plots of these learning graphs are not only applied for unraveling the bugs and search for hyper parameter, but also connect exceptionally well with the estimated quality of the sample.

Moreover in WGANs, the discriminator acts like a critic rather than a classifier. Here the word "critic" means that it is trying to calculate the Wasserstein gap between the distributions P and Q where P is the estimated probability distribution from the generator and Q is the real distribution. The discriminator is not outputting the probability but the distance. So there is no Sigmoid layer at the output layer. Output is taken out directly without any activation function. For Wasserstein distance to be computed, the discriminator needs to be trained more number of times than the generator. Usually it is a 1:5 ratio. This means that the learning process of discriminator is five times more compared to its counterpart. In contrast with the original GAN technique, the WGAN algorithm initiates the following actions on the data set:

1. When each gradient is updated on the critic functionality, the WGAN algorithm caps the hyper parameters such as weights to a lesser and fixed bounds.
2. WGAN Algorithm Applies an Advanced Cost Borrowed with EM Gap without any Logarithmic Dependency

Discriminator network is not directly acting as a monitor or critic but as an apprentice for computing and concluding the Wasserstein metric between the actual and the generated probability distributions. WGANs allow us to get the critic trained till some point of optimality. Then on the completion of this training, the WGAN model simply delivers the computes cost or loss function to the generator. Later, the generator can be trained as any other neural network without the need for managing and balancing the capability of both the generator and the critic. The quality of gradients used for empowering and training the generator depends on how better the discriminator acts as critic in its performance. The comparison of the gradients in GANs and WGANs is shown in Fig. 6 where there are linear gradients appearing in the case of WGANs.

Tests conducted using WGAN method show the following important advantages over the standard traditional GANs:

1. Stability related to optimizing of a process being refined.
2. Improved cost function between the converged output of generator and the expected quality of the sample.

Coming to the other side of WGANs, they are not so excellent in their performance. As per Arjovsky et al. [10], the original authors of the WGANs research suggested that the capping of weights is a dreadful way to impose a Lipschitz continuity constraint. The WGAN models still experience the training that is unstable and converge slowly in some cases after weight capping where the capping window is too large. WGANs also suffer from the gradients that vanish slowly when the capping is too small. But as per Cheng et al. [15], the authors have mentioned that there are some enhancements possible by absolutely compensating the weight capping with the new technique called as the gradient penalty to overcome the limitations of WGANs. With all the knowledge accumulated so far, a case study on data augmentation using WGANs is discussed in the next section.

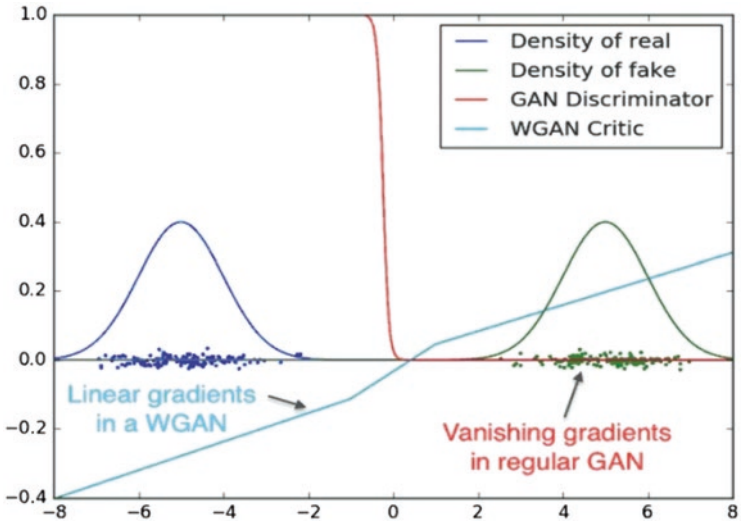


Fig. 6 Comparison of gradients in GANs and WGANs. (Courtesy: Margaret Maynard-Reid)

4 Data Augmentation Using WGANs: A Case Study

This section explains a chest X-Ray data set augmentation for COVID-19 detection using WGANs. It is proved that the WGAN estimated images are of far better quality than actual data obtained from reasoning test with existing COVID-19 analysis models. Use of WGANs in data augmentation can lead to an impressive and imponderous solving. The WGAN network in this case study is trained with two data sets from the X-ray repository.

1. The first one with the general or common and pneumonia images.
2. The second one with the general or common, pneumonia, and COVID-19 images.

By using Wasserstein distance between the produced and the real data, WGANs are capable of generating new and advanced X-ray images that are do not dependent on the image labels. These independent data generated would advance the accuracy of WGAN models thus resulting in the authentic classification of the COVID-19 and pneumonia cases excluding the common ones. Although there is an increased number of supervised deep learning models that have attained the encouraging results in the diagnostics of medical as well the agricultural [16] domains of imaging data, they need a heavy quantity of labeled data to study, generalize, and categorize them in order to categorize COVID-19, pneumonia, and normal flu with higher accuracy. But in the case of WGANs, they are capable of generating and estimating unknown area related data with no regard to the category of the image input data. Due to these reasons, an unsupervised data augmentation appeared where there are no labelled data available in the data set. Such a WGAN architecture with the generator and the critic is shown in Fig. 7.

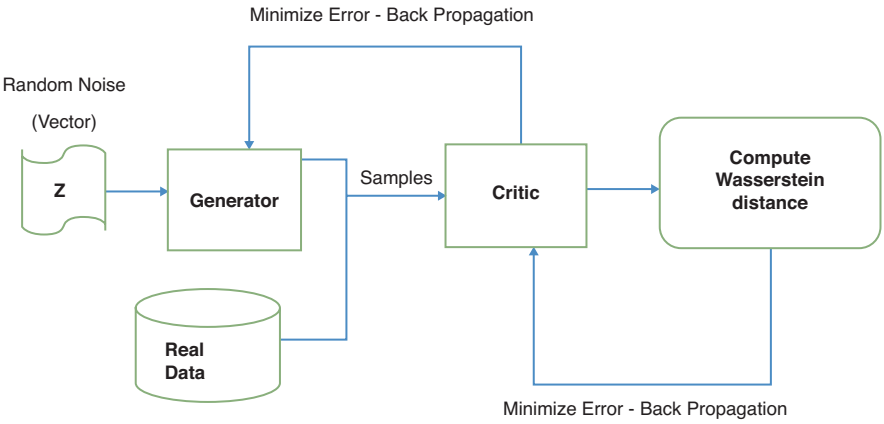


Fig. 7 WGAN Architecture for COVID-19 detection

As shown in Fig. 7, during every repetition i, generator GN takes White or Gaussian noise vector input zi and a set xi with actual learning image data. Next the input data images are encoded with the CNN layers and transformed into a low dimensionality depiction just before mixing these low dimensional abstraction of the data with the predetermined zi vector. This mixing occurs when input image data passes through a thick nonlinear hidden layer. It is done with an intention to use the complete image depiction with critic but to also obtain a lesser depiction of the image data that is input via GN. This is done to produce quality estimation and generalization of the quality image data. Two-way input to the generator (i.e., noise and training image data) also encourages the trained generator to make use of the image data from various categories and generate a deep range of image data to aggregate the required training data category. In this case, a 1 × 1 convolution method is used to help lower the number of channels in the WGAN model. After calculating the Wasserstein distance between the real and generated distributions, the loss is back propagated to both the critic and generator to minimize the error. A well-trained generator in WGAN learns the map function GN (z): z > =x from the latent space depictions z to the actual X-ray image data.

In general, the optimization [17] of the generator and the discriminator in GANs can be thought of as min max game as shown in Eq. 16:

$$\min_{GN}\max_{CR}\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log C\left(x\right)\right]+\mathbb{E}_{z\sim p_z(z)}\left[1-\log C\left(GN\left(z\right)\right)\right] \tag{16}$$

In this process, the generator during its training is made to learn how to lessen the accuracy of the critic CR's capability to differentiate actual and produced images with critic working for boosting the probability of assigning actual learning image data. While in the training phase, GN advances the process of generating more realistic image data while CR tries its level best to accurately recognize the difference between the actual and produced images. Generated and categorized images are shown in Figs. 8 and 9, respectively.

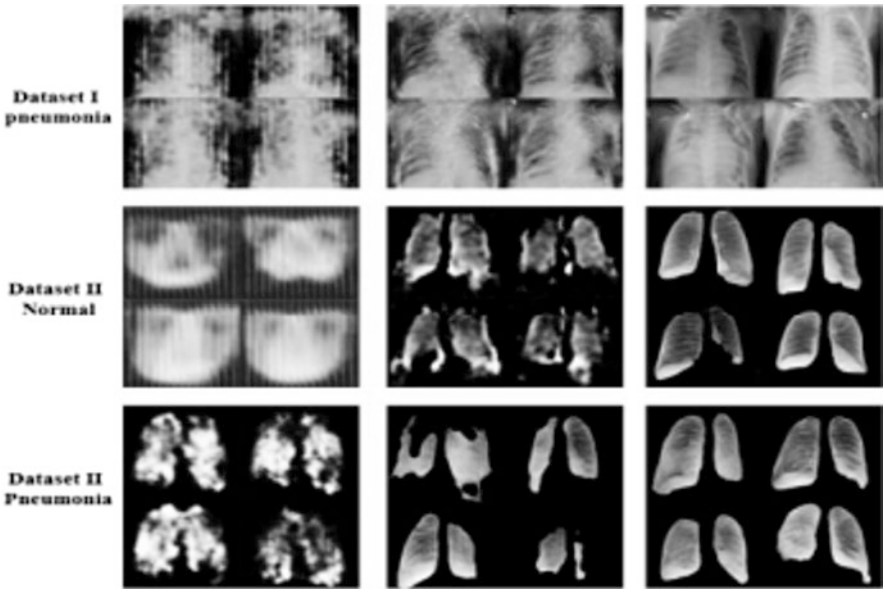


Fig. 8 Generated images. (Courtesy: Patrik Rogalla et al.)

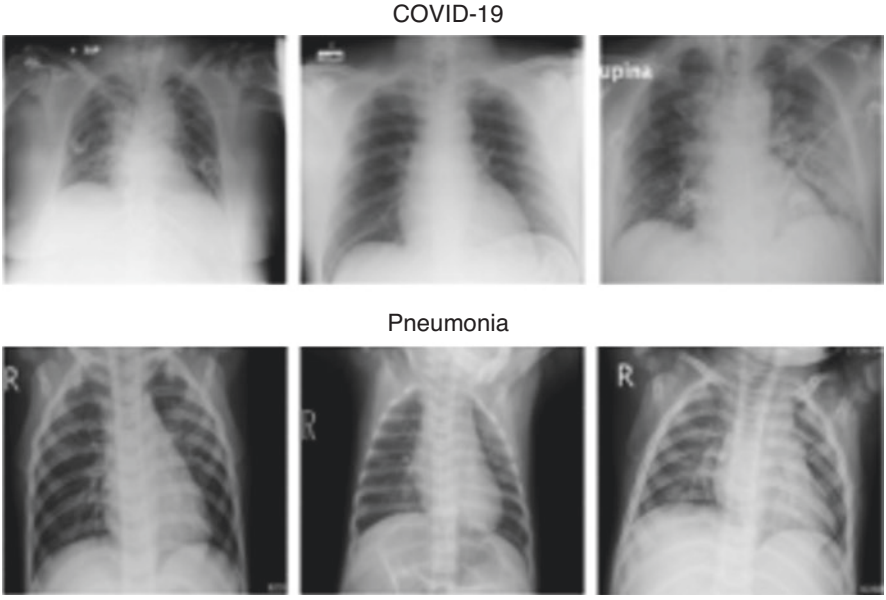


Fig. 9 Classified images. (Courtesy: Patrik Rogalla et al.)

Fig. 8 demonstrates the output of generator at the beginning, middle, and final phases from one direction to the other direction, respectively. For this, the generator is trained with both the data sets [17].

As shown in Fig. 9, note that there are markings on the top left part of the images related to COVID-19 cases. Similarly, a symbol R is located on the images related to pneumonia. These pneumonia-related images are persistent to the data images coming from the corresponding data set.

As shown in Fig. 9, note that there are markings on the top left part of the images related to COVID-19 cases. Similarly, a symbol R is located on the images related to pneumonia. These pneumonia-related images are persistent to the data images coming from the corresponding data set.

5 Conclusion and Future Scope

The chapter introduced WGANs as one of the data augmentation techniques. WGANs are an alternative to traditional GAN models. They provide the learning process of probability distributions in very large dimensional spaces. The chapter has done a comprehensive investigation regarding the Wasserstein or Earth Mover distance in contrast with the popular probability distances and divergences applied in order to learn the distributions. Dropping of mode issue in GANs is tremendously decreased in WGANs. The issues with the KL and JS divergences can be overcome with the introduction of Wasserstein or EM distance in generating and estimating the quality data. WGANs particularly avoid the learning issues related to GANs. The research on WGANs proved to be fruitful and can be shown that there can be an advancement in learning stability in the deep neural networks. Also the issues like mode collapse and convergence can be managed competently. These network models also deliver the substantial learning graphs that are useful not only for testing but also for searching the hyper parameters. Although WGANs provide the means for generating and estimating the best quality data, they still suffer with an unstable learning process and slower convergence process when the window cap is much higher. Vanishing gradients also pose issues when the window cap is too low. These fluctuating issues of gradients can be handled by compensating weight capping with what is known as gradient penalty. Gradient penalty is the smooth variation of the Lipschitz constraint that can be enforced on the critic's output. The ongoing and future research issues with WGANs include the following: (1) WGANs can be made even faster compared to the other data augmentation techniques, and (2) resolving the gradient issues efficiently by accommodating new and advanced constraints like Lipschitz.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. In *Advances in neural information processing systems* (pp. 2672–2680).
2. Donahue, C., McAuley, J., & Puckette, M. (2019). Adversarial audio synthesis. *In International conference on learning representations*.
3. https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/
4. https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
5. Banks, J., Carson, J., Nelson, B., & Nicol, D. (2009). *Discrete-event system simulation*. Prentice Hall.
6. https://www.probabilitycourse.com/.
7. Tolstikhin, I. O., Gelly, S., Bousquet, O., Simon-Gabriel, C. J., & Schölkopf, B. (2017). Adagan: Boosting generative models. In *Advances in neural information processing systems* (p. 30).
8. https://tinyurl.com/84rhrskb
9. https://lilianweng.github.io/posts/2017-08-20-gan/
10. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223).
11. Cheng, C., et al. (2020). Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis with unlabeled or insufficient labeled data. *Neurocomputing, 409*, 35–45.
12. Wei, X., Gong, B., Liu, Z., Lu, W., and Wang, L. (2018). Improving the improved training of Wasserstein GANs: A consistency term and its dual effect. In *International Conference on Learning Representation (ICLR)*.
13. Ponti, M., et al. (2017). A decision cognizant Kullback–Leibler divergence. *Pattern Recognition, 61*, 470–478.
14. Menendez, M. L., et al. (1997). The Jensen-Shannon divergence. *Journal of the Franklin Institute, 334*, 307–318.
15. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein GANs. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 5769–5779).
16. Jena, T., Rajesh, T. M., & Patil, M. (2019). Elitist TLBO for identification and verification of plant diseases. In *Socio-cultural inspired metaheuristics* (pp. 5769–5779). Springer.
17. Motamed, S., Rogalla, P., & Khalvati, F. (2021). Data augmentation using generative adversarial networks (GANs) for GAN-based detection of pneumonia and COVID-19 in chest X-ray images. *Informatics in Medicine Unlocked, 27*, 100779.

Image Segmentation in Medical Images by Using Semi-Supervised Methods

S. Selva Kumar, S. P. Siddique Ibrahim, and S. Kalaivani

1 Introduction

Many medical image-based clinical applications depend on accurate image segmentation. In current history, deep neural networks have been successful at segmenting images well, but they need a lot of annotated training data. For medical images, it is hard to get a lot of annotated examples because it is time-consuming and expensive to get medical specialists to annotate a lot of segmentation covers, requiring per-pixel records. Image augmentation has been proven to be a good and effective way to deal with this problem. Image augmentation is a collection of methods used to increase the amount and quality of training datasets for deep learning models. In medical imaging, augmentation transforms the images and the labels, making the training data look distorted. Transformations like rotations, reflections, and elastic deformations are often used in augmentation methods to make training images look like one training example. When neural networks are subjected to training with a substantial quantity of labeled samples, the connections between visual elements and segmentation become comprehended. The masks become more resistant to changes in the form and intensity of the objective as well as the structures that surround it. However, segmentation algorithms that have only been exposed to a limited number of annotated examples accomplish inefficiently on test images, including variances not detected throughout training. These differences in shape are

S. Selva Kumar (✉) · S. P. S. Ibrahim
School Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, Andhra Pradesh, India
e-mail: selvakumar.s@vitap.ac.in; siddique.ibrahim@vitap.ac.in

S. Kalaivani
School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, Tamil Nadu, India
e-mail: kalaivanis@vit.ac.in

A. Solanki, M. Naved (eds.), *GANs for Data Augmentation in Healthcare*,
https://doi.org/10.1007/978-3-031-43205-7_14

caused by differences in how populations are built, and differences in intensity are caused by differences in (1) image acquisition, (2) tissue properties and compositions, and (3) scanner protocols, particularly in Magnetic Resonance Imaging (MRI). The semi-supervised data augmentation algorithms have achieved high performance in segmentation over a huge amount of annotated data [1].

1.1 Semi-Supervised Learning

Semi-supervised learning (SSL) approaches use unlabeled data to supplement the restricted labeled data used for training. The primary objective is to prevent overfitting while also making training more consistent through the utilization of unlabeled data. Let x represent the image and y represent the image pixel label map. The training Dataset D, which is used for training, includes both the label maps and the image pairs, $D = \{X, Y\}$ where $X = \{x_i, i = 1,2,3\ldots n\}$, and $Y = \{y_i, i = 1,2,3\ldots n\}$, where i represents the image index. Suppose there are two datasets in our hands, an unlabeled dataset $D_u = \{X_u, Y_u\}$ and a labeled dataset $D_L = \{X_L, Y_L\}$. The label maps Y_L are identified and usually arise from physical segmentation by specialists on images X_L, while Y_U are unknown. Then build a model for manual segmentation where label map y from image x with a network parameterized by Θ. In supervised learning, minimize the loss (L) with stochastic gradient descent (SGD) and achieve the optimized segmentation. The unlabeled dataset DU was introduced in semi-supervised learning to achieve optimization:

$$\min_{\theta,\mathrm{YU}} \mathrm{L}\left(\Theta,\mathrm{YU}\right) = \sum_{i\in L}^{n}\sum_{j} \log p\left(y_{i,j}|,x_i|,\Theta\right)\sum_{i\in U}\sum_{j} \log p\left(y_{i,j}|x_i,\Theta\right). \tag{1}$$

The cross-entropy for unlabeled data are mentioned in the second term in Eq. 1 and λ has represented the weights of unlabeled term data. The network factors (Θ) and the unidentified label maps YU need to be considered when optimizing the loss function.

The SSL is classified into the small categories shown in Fig. 1.

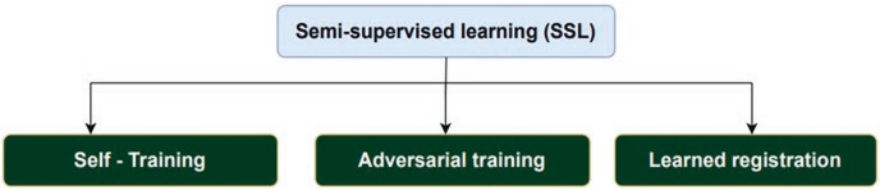


Fig. 1 Categories of semi-supervised learning (SSL) methods

2 Self-Training Semi-Supervised Learning (SSL) Methods

In self-training, the learner labels examples on their own as part of a semi-supervised learning procedure that has not been labeled and retraining itself on an expanded set of examples that have been labeled. A labeled data collection is used to train a "base learner" in the self-training process. Then, it tries repeatedly to label the instances in the unlabeled set about which it is most confident. Then, it adds these self-labeled examples to its labeled training set. As a result of the insufficiency of the labeled set for learning, it is impossible to avoid misclassifying some unlabeled data.

2.1 Network-Based Cardiac MR Image Segmentation

The network-based cardiac MR image segmentation method [2] obtained the loss function by alternatively updating the values of both network parameters (Θ) and the unlabeled dataset (Y_U) shown in Fig. 2. The early values are attained by training the network for several epochs using only labeled (Y_L) maps. First, the network parameter values are fixed $\hat{\Theta}$, then it optimizes the unlabeled image term loss function. In this step, segmentation is done on the unlabeled images based on the current network. If the unlabeled data value (Y_U) is fixed, then it estimates the network parameters (Θ). These actions are performed by training on both the labeled training data and the estimated segmentations (Y_U) to update the network's parameters (Table 1).

The strategy incorporated unlabeled data, which improved segmentation performance, particularly with a minimal training set. Segmenting a single subject takes

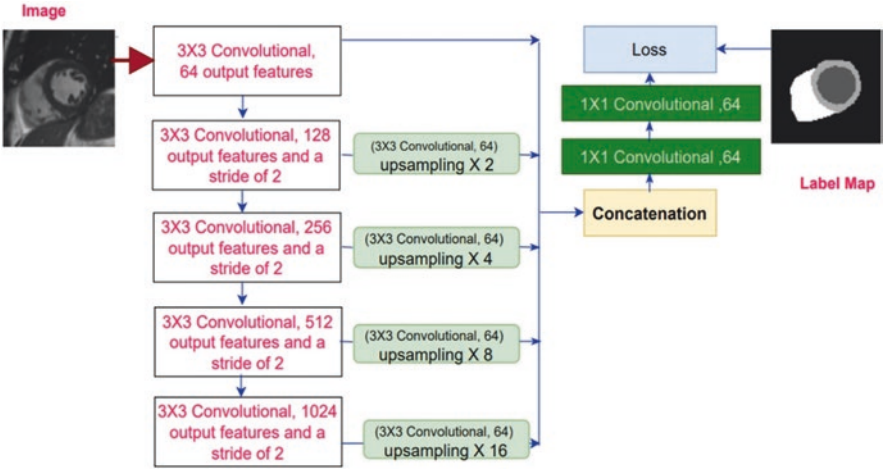


Fig. 2 Fully convolutional network architecture

Table 1 Network-based Cardiac MR image segmentation procedure

Step 1: Utilize the network to determine the softmax probability and a conditional random field (CRF) to evaluate the probability map for a more precise segmentation (unlabeled data segmentation is improved with CRF)
Step 2: Similar to supervised learning, SGD can be used to optimize the cross-entropy loss function
Step 3: Do steps 1 and 2 alternatively until the network parameters improved significantly when the segmentations were updated, and vice versa

just a few seconds when the network has been properly trained. The main drawback of this method is that if a bias or fault (over- or under-segmentation) happens, the network will learn a mistake in the first segmentation of the unlabeled input.

2.2 Self-Paced and Self-Consistent Co-training Method

The minimization of entropy is one method that has been suggested for enhancing knowledge in semi-supervised classification problems [3]. The self-paced and self-consistent co-training method extends the concept of semi-supervised entropy regularization and self-paced learning [4] and [5]. This method considers three dissimilar losses: (1) supervised loss (pixel-wise), (2) self-placed co-training loss, and (3) self-consistency loss.

2.2.1 Self-Paced Learning

The self-paced learning model is erudite during training by accumulating more complicated examples. In the typical self-paced learning model, a self-paced regularize is included in calculating the weights applied to each instance in the learning goal at a specified learning speed [6]. In this self-paced method, high-confidence parts of unclear images are looked at first, followed by those with less confidence. The main objective of the self-paced co-training is to minimize the segmentation loss by using the cross-entropy loss. The self-paced co-training method is based on the Jensen-Shannon divergence (JSD). The standard JSD method is used to measure the inter-model agreement. The Jensen-Shannon divergence (JSD) algorithm dynamically modifies the significance of individual pixels during the training process of various segmentation networks. Furthermore, when labeled data is narrow, then utilize a self-consistency cost based on chronological assembling to standardize individual models' training further and boost execution. The self-paced and self-consistent co-training method has a drawback. The task at hand necessitates the operation of numerous segmentation networks, thereby rendering the computational requirements more challenging to fulfill. This constraint can be overcome by using parallel calculation techniques to increase speed training and implication, but it could also be solved by making a single model that uses the information from

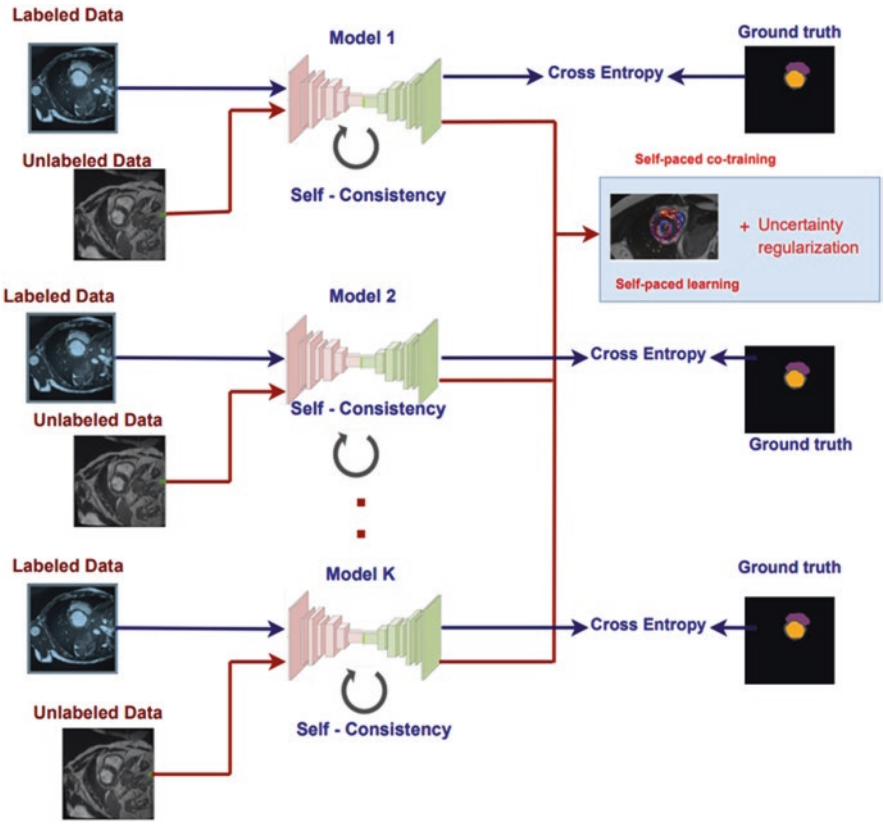


Fig. 3 Self-paced and consistent co-training design for semi-supervised segmentation

several models. Another big problem with this method is that you have to find a balance between different loss terms that may fight with each other throughout the training (Fig. 3).

3 Adversarial Training Method

In computer security, an "adversary" attempts to trick or mislead a machine learning model. Adversarial machine learning is another approach that may be used to undermine the efficacy of any machine learning model by tricking it into using incorrect data. Adversarial training, in which a network is trained with examples of attacks from the other side, is one of the few ways to protect against attacks from the other side that can stand up to a strong attack [7].

In biomedical image analysis, semantic segmentation is one of the most important problems. An important question is how to use unlabeled images to train good

segmentation models. It is not a new idea to use unlabeled and labeled data to train a learning model. It added an unsupervised learning task to help train a neural network in a supervised way. Both unsupervised and supervised learning tasks use the same intermediate layers. Despite the fact that unsupervised and supervised learning is intended to accomplish separate things, the unsupervised learning component can on occasion lend a hand to the supervised learning component by way of the mutual model parameters. Using annotated and unannotated data would be ideal for serving the same goal. The big problem is that there are no real facts for unlabeled data, so back-propagation faults cannot be directly calculated after the forward pass. An adversarial network is used to train a deep neural network to calculate estimated faults for unannotated data [8].

3.1 Deep Adversarial Network (DAN)

The deep adversarial network (DAN) method is used in image segmentation for both labeled and unlabeled medical images [9]. The DAN contains two network models: (1) an evaluation network (EN) and (2) a segmentation network (SN). The segmentation network conducts the segmentation process, and the evaluation network determines the image segmentation quality. The architecture of the DAN model is represented in Fig. 4. First, SN is trained with labeled images along with ground truth images. Then, EN has been trained to assign dissimilar scores to labeled and unlabeled image segmentations according to its training. The

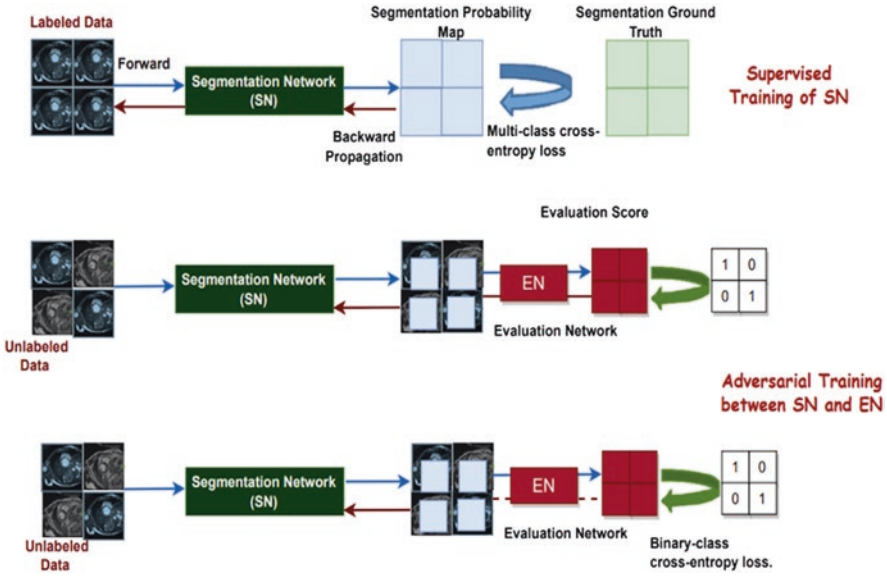


Fig. 4 The architecture of the deep adversarial network (DAN)

enhancement of segmentation quality in SN training was achieved by utilizing the feature map of the image learned in EN.

The information given to EN as input is critical to the entire adversarial training system. The segmentation probability maps provide EN with the opportunity to investigate helpful morphological aspects of the segmented biological objects and evaluate the segmentation quality, which might be a simple form for EN to use as input. Combining segmentation probability maps with the input image corresponding to them is a strategy to build input for EN that is more effective than other methods. The purpose of the evaluation network (EN) is to assess the quality of segmentation by analyzing the correlations between the input image and the resulting segmentation.

There are two different methods for combining the segmentation probability map and the input image. First, put them together directly or turn them into two feature maps before concatenating them. Since EN uses different model parameters to process evidence from the segmentation files and the input image, it is feasible that the facts from the raw image data are used to make decisions. The deep adversarial network can efficiently use unlabeled image data to train biomedical image segmentation neural networks for improved simplification and resilience.

3.2 SGNet Image Segmentation

The SGNet used an adversarial network learning approach to unlabeled data [10]. This approach in GAN consists of two segments called the generator G and the discriminator D with a fully convolutional design similar to that of the widely used U-Net. This approach proposed a new semi-supervised method that utilized the convolutional deep learning method for segmenting the medical OCT B images. The unlabeled images train through an adversarial network along with supervised training data. The discriminator network uses unlabeled images and it is very effective in semi-supervised loss. The segmentation network act as an encoder of the segmentation network and it trained the images adversarial and semi-supervised manner. This work is an attempt to train the unlabeled image by using adversarial learning and balancing both cross-entropy loss and semi-supervised loss. Figure 5 depicts the working principles of SGNet segmentation architecture.

3.3 Multi-path and Progressive Upscaling GAN-Based Method

The photo-realistic single image super-resolution using a generative adversarial network (SRGAN) is used to remove low features on diverse scales of the medical images [10, 11]. The multipath method images upscale by using two steps, in the first step extract the shallow features after that extracts deep features in the images these two steps are achieved by using ResNet34 architecture. In this architectural

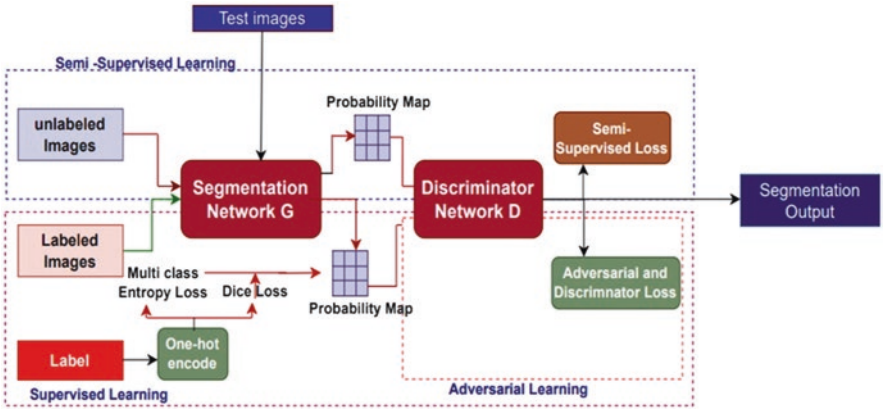


Fig. 5 The architecture of the SGNet segmentation method

approach, the initial extraction of superficial attributes in magnetic resonance (MR) images is achieved through the utilization of a low-resolution rendition of the distinctive high-resolution image as an input. The two blocks of the convolutional layer are constructed and the first block act as batch normalization and activate ReLU the next block is again the batch normalization layer. In the deep feature extraction phase, this phase low-resolution images are upscaled by 2× the features extracted from the upscaled version of images. The three loss functions considered in this work are content loss, generator loss, and mean square loss. This method improves the overall quality of low-resolution images into high-resolution MR images.

4 Conclusion

Due to the exertion of compiling large-scale labeled datasets, the clinical application of effective deep-learning methods for medical image investigation is limited. This chapter discussed some novel methods to handle both labeled and unlabeled medical images for segmentation. The semi-supervised learning methods are very limited in literature but they are more effective and handled the medical images.

References

1. Chaitanya, K., et al. (2021). Semi-supervised task-driven data augmentation for medical image segmentation. *Medical Image Analysis, 68*, 101934. https://doi.org/10.1016/j.media.2020.101934
2. Bai, W., et al. (2017). *Semi-supervised learning for network-based Cardiac MR image segmentation*. Lecture notes in computer science (pp. 253–260), https://doi.org/10.1007/978-3-319-66185-8_29.

3. Grandvalet, Y., & Bengio, Y. (2004). Semi-supervised learning by entropy minimization. In *NIPS'04: Proceedings of the 17th international conference on neural information processing systems* (pp. 529–536).
4. Wang, P., Peng, J., Pedersoli, M., Zhou, Y., Zhang, C., & Desrosiers, C. (2021). Self-paced and self-consistent co-training for semi-supervised image segmentation. *Medical Image Analysis, 73*, 102146. https://doi.org/10.1016/j.media.2021.102146
5. Zou, Y., Yu, Z., Kumar, B. V. K., & Wang, J. (2018). Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 289–305). https://doi.org/10.1007/978-3-030-01219-9_18.
6. Qiao, S., Shen, W., Zhang, Z., Wang, B., & Yuille, A. (2018). Deep Co-training for semi-supervised image recognition. In *Computer vision – ECCV* (pp. 142–159) https://doi.org/10.1007/978-3-030-01267-0_9.
7. Park, S., & So, J. (2020). On the effectiveness of adversarial training in defending against adversarial example attacks for image classification. *Applied Sciences, 10*(22), 8079. https://doi.org/10.3390/app10228079
8. Souly, N., Spampinato, C., & Shah, M. (2017). Semi supervised semantic segmentation using generative adversarial network. In *2017 IEEE International Conference on Computer Vision (ICCV)* https://doi.org/10.1109/iccv.2017.606.
9. Zhang, Y., Yang, L., Chen, J., Fredericksen, M., Hughes, D. P., & Chen, D. Z. (2017). Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017* (pp. 408–416), doi: https://doi.org/10.1007/978-3-319-66179-7_47.
10. Liu, X., et al. (2019). Semi-supervised automatic segmentation of layer and fluid region in retinal optical coherence tomography images using adversarial learning. *IEEE Access, 7*, 3046–3061. https://doi.org/10.1109/access.2018.2889321
11. Ahmad, W., Ali, H., Shah, Z., & Azmat, S. (2022). A new generative adversarial network for medical images super resolution. *Scientific Reports, 12*(1). https://doi.org/10.1038/s41598-022-13658-4