Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6, 2015-2024
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i6.2372

Improving losses & accuracy through design of deep convolutional generative adversarial network (DCGAN) for plant disease detection tasks

Pooja Sharma[1*], Ajay Khunteta[2]
[1]Poornima University, Jaipur, India; er9.pooja@gmail.com (P.S.)
[2]Dept of CSE, Poornima University, Jaipur, India; ajay.khunteta@poornima.edu.in (A.K.).

Abstract: Rice leaf diseases are a significant issue that adversely impact rice production in India. Identifying these diseases manually is labour-intensive and prone to delays, often resulting in substantial crop losses for farmers. Therefore, the need for an automated system for early detection of plant diseases is critical. Recent advancements in machine learning, computer vision, and deep learning have paved the way for classification models capable of automatically identifying these diseases. However, the challenge lies in obtaining a sufficiently large and diverse image dataset to effectively train deep learning models. In this paper, we address this limitation by employing advanced data augmentation techniques, including Deep Convolutional Generative Adversarial Networks (DCGANs), to generate synthetic images that expand the dataset of rice leaf diseases. By integrating these synthetic images with real images, a new Convolutional Neural Network (CNN) architecture is proposed, which offers improved generalization capabilities. The performance of the classification model is evaluated with and without the DCGAN-generated images. The results demonstrate that the inclusion of synthetic images significantly enhances accuracy, as the enlarged dataset better represents real-world conditions. This approach provides a promising solution for more effective rice disease identification, offering higher precision in real-time scenarios.

Keywords: *Automated diagnosis, Classification, CNN, Computer vision, Deep Learning, Generative adversarial networks, Plant diseases, Rice leaf diseases, Smart farming, Synthetic images.*

1. Introduction

Food safety is a critical issue in balancing agricultural production with increasing demand [1]. In India, the agricultural sector contributes 17.5% to the country's GDP, with production levels rising annually [2]. Rice, a staple crop and one of the most cultivated cereals in India, plays a pivotal role in this growth. It is primarily a kharif crop, thriving in temperatures above 25°C and requiring over 100 cm of rainfall. In regions with insufficient rainfall, irrigation systems support rice cultivation. However, fluctuations in production are common; for instance, rice production in 2009-10 dropped to 89.13 million tons from the previous year's 99.8 million tons [3].

To address this, improving agricultural productivity is essential, especially given India's relatively low average yield compared to other major rice-producing countries [4]. Visible effects of plant diseases, referred to as symptoms, are critical indicators in diagnosing issues. These symptoms include changes in color, shape, and overall plant function, which are caused by either abiotic (non-infectious) or biotic (infectious) factors. Abiotic diseases stem from external factors, such as nutrient deficiencies or soil problems, and do not spread between plants. In contrast, biotic diseases are caused by pathogens like fungi, bacteria, viruses, and nematodes, which can spread and visibly affect leaves, roots, seeds, fruits, and stems.

* Correspondence: er9.pooja@gmail.com

This research focuses on the identification and classification of rice diseases using advanced technologies. Domain experts have been consulted for a deeper understanding of common rice diseases [5]. Precision farming, leveraging technological advancements such as remote sensing and GIS, helps farmers optimize inputs like water, pesticides, and fertilizers. Recent developments in computer vision and deep learning techniques, particularly Convolutional Neural Networks (CNNs), offer promising solutions for automatic image classification, addressing complex agricultural challenges [6,7].

1.1. Related Work

Convolutional Neural Networks (CNN) are widely used in the development of image classification and object detection models. Phadikar et al. [8] proposed a method for identifying rice diseases using pattern detection techniques, specifically the Self-Organizing Map (SOM) neural network. The authors introduced a zoom-in algorithm that renders image features through a simple, computationally efficient process. Sanyal, P., and Patel, S.C. [9] focused on detecting patterns of two rice plant diseases using a Multi-Layer Perceptron (MLP) classifier. Their validated analysis showed that 89.26% of pixels were accurately organized.

A. Smith, J.S. [10] proposed a method for identifying plant pathogens based on visible signs of disease in color photographs. The authors used a Support Vector Machine (SVM) to analyze 117 images of cotton plants, considering features like gray levels, connectivity, and texture. H. Al-Hiary and S. Bani-Ahmad [11] suggested a rapid and precise method for diagnosing and classifying plant-borne pathogens by integrating K-means clustering with Neural Networks (NNs). Bashir and Sabah [12] introduced a method for remote disease diagnosis in *Malus Domestica* using image analysis and K-means clustering.

Neumann et al. [13] proposed an approach for cell disease detection using core-specific features, achieving highly accurate predictions on mobile devices with low-cost computing. Sachin D. Khirade [14] developed a method for plant disease diagnosis using artificial neural networks (ANN), classifying diseases from RGB leaf images. Various methods such as Feature Mapping, backpropagation algorithms, and SVMs were effectively employed.

P. Mohanty et al. [15] developed a deep learning model on a public dataset and created a mobile application for real-time plant disease detection. Sladojevic et al. [16] proposed a model capable of distinguishing 13 different plant pathogens using a Deep Neural Network, offering potential for global application in leaf disease detection. Grinblat et al. [17] suggested an in-depth study of plant identification using morphological vein patterns, while Barbedo et al. [18] focused on identifying multiple plant diseases using digital image processing, addressing the gap between current image-based diagnostics and real-world needs. Alvaro F. Fuentes [19] introduced a three-step process that improved the identification of plant diseases by focusing on different plant parts (stems, leaves, and fruits), achieving approximately 96% accuracy.

1.2. Problem Statement

Deep learning experiments, particularly those involving Convolutional Neural Networks (CNNs), require large datasets to effectively train models. In cases where large datasets are unavailable, model accuracy and precision often decline. To address this challenge, a Deep Convolutional Generative Adversarial Network (DCGAN) is utilized to generate synthetic images. The model's accuracy is then evaluated using a combined dataset of original and synthetic images.

This paper is organized as follows:

- The architecture used to generate synthetic images using DCGAN.
- The DCGAN architecture, which includes a Generator network to create images and a Discriminator network to differentiate between generated and original images.
- The CNN model used to classify diseased images after integrating synthetic images with the original dataset.

ISSN: 2576-8484
Vol. 8, No. 6: 2015-2024, 2024
DOI: 10.55214/25768484.v8i6.2372

- Evaluation of the deep learning model's performance using standard metrics after training with the combined dataset.

2. Dataset & Methodology

2.1. Dataset

In this paper, the Rice Image Dataset [20] was selected, which comprises a total of 5,932 images across various categories of rice diseases, including Bacterial Blight, Blast, Brown Spot, and Tungro. To address the challenge of limited data, basic data augmentation techniques were applied to enhance the diversity of the training set. These techniques included transformations such as image rotation, flipping, scaling, and cropping. By artificially expanding the dataset, the model was provided with a greater variety of image patterns, which helped reduce overfitting, improve generalization, and enhance the precision of the deep learning model. This approach ensured that the model could better recognize and classify different disease types, even with variations in image orientation or scale.

Figure 1.
(A)Bacterial blight, (B)Blast, (C)Brown Spot and(D)Tungro.

3. Methods

3.1. Generative Adversarial Network (Gan)-

The goal for GAN is to model "how the data looks like (density estimation)" and generate new data based on what it has learned. GANs are unsupervised learning algorithms with loss as a supervised learning component that becomes part of the training. In 2014, Ian Goodfellow introduced a deep neural network architecture that leverages unsupervised machine learning to produce data [21,22]. A generative model's specialized frameworks are called GANs. When given a set of sample images, such as X1, X2 a model learns the data distribution (p_{data}) in order to produce new sample. It is made up of the discriminator (D) and the generator networks, which were trained concurrently (G). D makes a distinction between the actual and fraudulent images it has obtained. It receives an input of I and produces D, that is called as probability of original sample (i). On the other hand Generator module create new samples by mechanism of synthesis input j1, j2..,jn from a kind of uniform distribution p(i) and pairing with G(j) to create new image dimension p(g).Generator tries to achieve f(g)=f.(data)[35].

To minimize 1 D(G(j)) or maximize D(G(j)), the generator is trained, during training, G increases its ability to create more realistic images. D increases its ability to distinguish between real and artificial visuals.[35]

The generator produces a 128x128x1 sized image by utilizing a 100-dimensional input vector containing random values drawn from a uniform distribution. The network architecture includes four fractional-stride convolutions with 4x4 sized kernels to up-sample the image, along with a fully connected layer scaled to 4x4x128. appending zeros in the pixels expands our output of each fractional stride convolution. At every layer to handle gradient vanishing problem batch normalization is added at every layer, including the output layer, to normalize the output across the mini batch. This normalization prevents the generator from failing all samples to a single point and helps stabilize the learning process of the DCGAN. The output layer employs the Tanh activation function, while all other layer's use ReLU.

The discriminator network receives 128x128x1 images as input and produces result regarding the validity of the images. It has five completely connected layers with and four convolutions. Stride convolutions are used to reduce the spatial dimensionality. The likelihood probability score of images is output using the sigmoid function as [0, 1]. Additionally, every layer—aside from the second and the output layers—uses the dropout technique to avoid the over-fitting problem.

3.2. Convolutional Neural Networks (Cnn)

Neural Network that has multiple convolutional layers and used for image processing, segmentation, classification and for data, which is auto corrected, is known to be Convolutional Neural Network (CNN), CNN is created using many layers, main layers of CNN are: The feature extraction layer i.e. Convolutional layer. When an image is passed as input to a CNN, a specific layer in the network detects the existence of a predefined set of features within the image. Each layer in the CNN can be seen as a filter applied to the image, where this layer performs a convolution operation between the feature and the scanned image. Next layer is called sub sampling or pooling layer. This layer is a mediator of two layers to perform pooling features from one layer and transfer to another without affecting their properties and reducing the size of input image [23,24].

To replace all values which are negative by zeros taken in input, performed by RELU which is Rectified Linear Unit function works as an activation function. Its function is defined as:

$$\mathrm{ReLU}(x)=\max(0,x) \qquad (1)$$

Activation Layers Activation Layers are used in-between each nonlinear function layer to activate the layers such as ReLU and Convolutional layer. Dropout layer regularizes and prevents overfitting through increasing testing accuracy based on training accuracy is Dropout Layer. It disconnects the inputs by probability p and improves the further layers. The last layer in CNN architecture is fully connected layer. Softmax / Logistic Layer Softmax or Logistic layer is used as binary classification and multi-classification and used just after Fully Connected layer. Batch normalization Batch normalization is the processes of normalization within the activation layer batch including subtraction of mean and deviation by the standard deviation. To make Neural Network more predictive it is required to minimize or remove prediction error, it is simply as the errors are loss, that is what Loss function works and calculates gradients. Loss functions update the weights of the network and Network trained. SoftMax or Logistic layer used separate cross entropy for binary and multi-class classification as:

Binary Cross entropy is used for binary classification process by passing output value through *sigmoid* activation function as:

$$L = -(y_i \log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \qquad (2)$$

Categorical Cross entropy is used for a multi- class classification process by passing output value through *SoftMax* activation as:

$$L = \sum_{j=1}^{M} y_j \log\left(\hat{y}_j\right) \qquad (3)$$

CNN optimizer calculating the gradients for each training examples is not practical, as it will lag in prediction accuracy. To optimize the process various methods are used, from that most used are: Stochastic Gradient Descent: Using batches at a time or randomly selecting examples on each pass. RMSprop: Gradients of all examples are collected in a fixed window and avoid collecting all for moment. Adam: It calculates the current gradients based on past gradients by using momentum by adding fractions from previous gradients.

3.3. Experiment & Analysis

In the proposed model for rice disease identification, the training and test datasets were first augmented using the Keras Image Data Generator. The Rice Image Dataset contains images of four disease types. To further enhance the dataset, a Deep Convolutional Generative Adversarial Network (DCGAN) was employed to generate synthetic images for two specific disease types: Blast and Tungro. Initially, the dataset comprised 5,932 images, but the application of DCGAN, trained over 1,000 epochs, generated an additional 2,000 synthetic images (1,000 images for each disease).

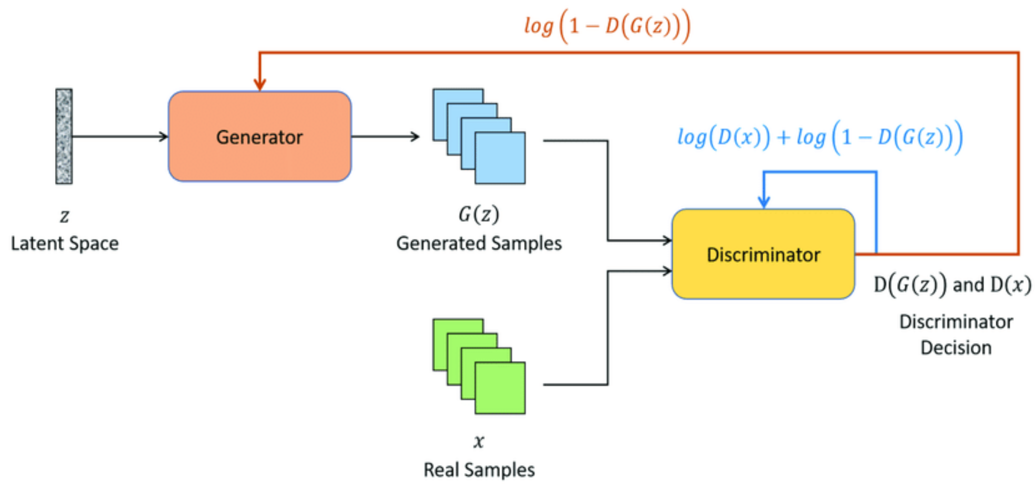

Figure 2.
DCGAN architecture.

For the DCGAN training, key parameters such as D(G(j1)), D(G(j2)), D(i), Loss D, and Loss G were monitored and plotted against the number of epochs to analyze the training process for both the generator and discriminator networks. Figure 3 illustrates the performance comparison between the real images of Blast and Tungro from the original dataset and the synthetic images generated by the DCGAN. The results indicate that as the number of epochs increases, both networks achieve greater stability.

For disease classification, a novel Convolutional Neural Network (CNN) architecture was utilized. The architecture comprises five convolutional layers, with Batch Normalization added to mitigate overfitting, and a Dropout rate of 0.25 to enhance test accuracy. The output layer employs the Softmax activation function. The model was fine-tuned with hyperparameters such as 25 epochs and a learning rate of 0.0001. Additionally, the model's performance was evaluated using three CNN optimizers: Adam, SGD, and RMSprop. Accuracy and Top-5 accuracy were used as evaluation metrics, with Categorical Cross Entropy as the loss function to assess the difference between actual and predicted classes.

The following steps were carried out:

1. The original Rice Image Dataset was augmented and used to train the CNN model, and the model's accuracy was evaluated.
2. A new dataset consisting of synthetic images of Blast and Tungro, generated by DCGAN, was used for data augmentation. This augmented dataset was then employed for further model training.

The performance metrics of the proposed CNN model and comparative analyses are detailed in the results section. The Keras deep learning library was used for model training, and both the DCGAN and CNN models were trained on Google Colab.

4. Results & Discussions

The proposed DCGAN model for generating synthetic data in the form of Blast and Tungro disease images has been evaluated to improve CNN classification performance. The generated images are mixed with the original dataset to increase the diversity of the training data. By utilizing DCGAN-generated synthetic data, the CNN model's training process can be enriched, leading to improved generalization, particularly for real-world test cases.

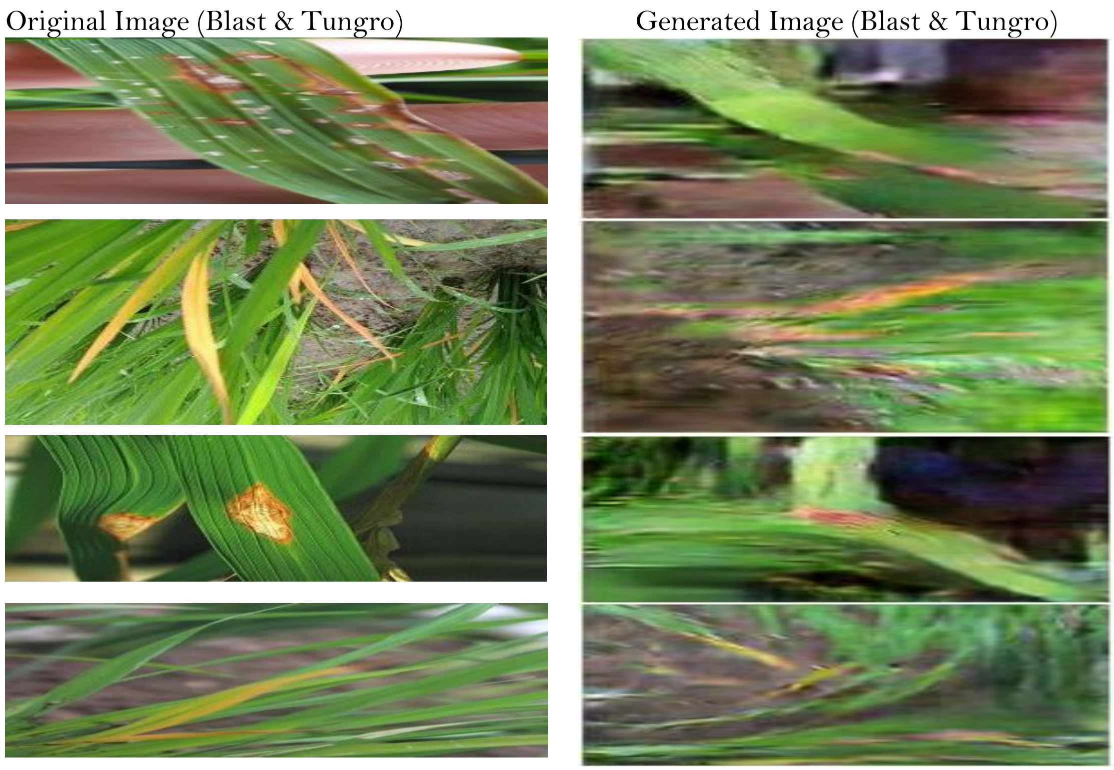

Figure 3.
Generated Images of Blast & Tungro through DCGAN.

The GAN network attempts to optimize the loss function V(D,G) given below[22]

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \qquad (6)$$

The use of generator which is here G(z) is used to convert noise z into which is input into some kind of data, as images. The role of discriminator D(x) is to discriminate and output the probability of to identify input coming is from real images or not. This is achieved by using log-likelihood function of D(x) and 1-D (z) that is defined in objective function. The performance of DGCAN is assessed using the average testing accuracy value. For each category, additional metrics like accuracy, F1-score, recall, precision computed & analyzed.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 2015-2024, 2024
DOI: 10.55214/25768484.v8i6.2372

Figure 4.
Epoch analysis (1-1000) for blast.

Figure 5.
Epoch analysis (1-1000) for Tungro.

Table 1.
CNN model accuracy without GAN.

Method	Optimizer	Loss function	Epoch	Validation accuracy
CNN (From Scratch)	SGD	categorical_crossentropy	25	98.75

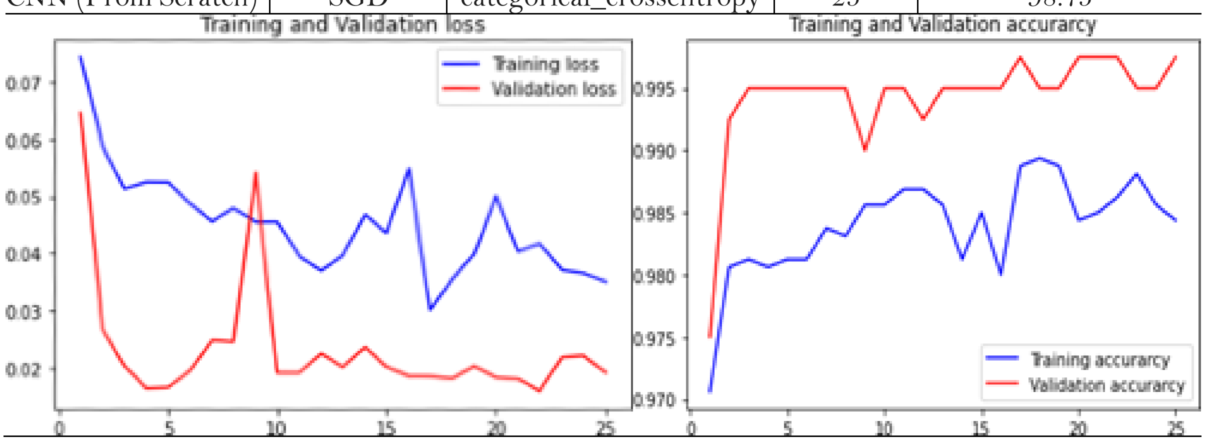

Table 2.
Accuracy of CNN with DCGAN.

Method	Optimizer	Loss function	Epoch	Validation accuracy
CNN (From scratch)	SGD	Categorical cross entropy	25	99.75

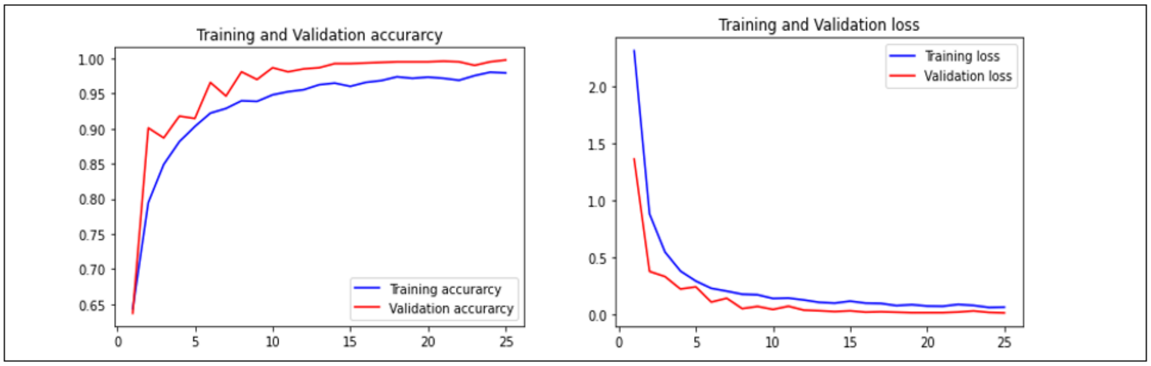

Figure 6.
CNN(DCGAN) training & validation loss.

5. Discussions

1. CNN with and without GAN Integration: Table 1 shows that the baseline CNN model, trained from scratch without synthetic data, achieved a validation accuracy of 98.75%. However, as shown in Table 2, after introducing DCGAN-generated images, the validation accuracy rose to 99.75%. This increase in accuracy underscores the advantage of using synthetic images generated by DCGAN, suggesting that the model was better equipped to generalize to unseen data.
2. Training and Validation Metrics: The training and validation loss and accuracy plots (shown in the figures) provide insights into the model's learning process. The validation loss consistently tracks the training loss, indicating that the model is not significantly overfitting. Over the course of 25 epochs, the validation accuracy remained stable and high, reaching close to 99.5%. The training loss steadily decreased over time, with validation loss showing more fluctuations but overall following the same downward trend. The proximity of the two curves indicates minimal overfitting.
3. DCGAN Epoch Analysis: Figures depicting the quality of DCGAN-generated images for Blast and Tungro diseases across different epoch values (200, 500, 1000) indicate that as the number of epochs increases, the quality of the generated images improves. This is particularly noticeable for epoch values of 1000, where the synthetic images are almost indistinguishable from real ones. The visual improvement suggests that the generator network in the DCGAN becomes more adept at producing realistic images with more training iterations.
4. Loss Function Behaviour: The generator and discriminator losses reveal an interesting trend. Initially, the generator loss increases, which might indicate that it struggles to fool the discriminator. However, as the epochs increase, the discriminator loss decreases, eventually reaching a minimum at 1000 epochs. This is a common observation in GAN-based models where, over time, the generator becomes more effective at producing realistic images, while the discriminator becomes less able to distinguish between real and fake images.
5. Quantitative Metrics and Performance: The research highlights various metrics for assessing the model's performance. In addition to accuracy, precision, recall, and F1-score are computed for both the real and synthetic datasets. The CNN model trained with DCGAN images outperforms the baseline in almost every metric. The increase in accuracy and other performance metrics can be attributed to the enlarged and diversified dataset created by adding synthetic images to the training set.

6. Training and Validation Loss Comparison: The training and validation loss curves for the CNN model with DCGAN reveal that the model is learning effectively without major signs of overfitting. Both the training and validation losses decrease steadily throughout the epochs. The fact that the loss curves for both datasets—real and augmented with synthetic images—are very similar further validates the effectiveness of DCGAN in generating meaningful data for the training process.

The results presented in this analysis highlight the power of synthetic data generation using DCGAN, especially in scenarios where the dataset is small or imbalanced. This study lays a foundation for future research into the benefits of GANs in generating data across various domains, including agriculture, medical imaging, and other industries requiring image classification. Future research could explore different GAN architectures, such as Wasserstein GANs (WGANs), to see if further improvements in the quality of synthetic data and model performance can be achieved. In conclusion, by utilizing DCGAN-generated images, the CNN model not only increased in accuracy but also became more robust and capable of handling diverse real-world data. The successful application of DCGAN in this study suggests that such synthetic data generation techniques could play a vital role in improving AI models' performance in areas where data is limited or difficult to obtain.

6. Conclusions and Future Plans

In this study, we explored the efficacy of employing Deep Convolutional Generative Adversarial Networks (DCGAN) for enhancing the accuracy of plant disease classification, specifically for rice diseases. By generating synthetic images through DCGAN, we successfully augmented the original dataset, thus significantly increasing the total number of training samples. This approach not only addressed the challenge of limited data availability but also ensured that the newly generated images closely resembled real-world data, thereby improving the generalization capabilities of the CNN model.

The comparative analysis between the models trained with and without DCGAN-generated images revealed a marked improvement in classification accuracy when synthetic images were included. The augmented dataset allowed the CNN to learn more diverse features, thereby equipping it with the ability to recognize plant diseases with higher precision in real-time scenarios. The stability of training and validation loss metrics across epochs further validated the robustness of the proposed approach, indicating minimal overfitting and effective learning dynamics.

Moreover, the evaluation of generator and discriminator losses highlighted the adaptive learning process of the DCGAN framework, where an increase in generator loss coincided with a decrease in discriminator loss, stabilizing at the 1000-epoch mark. This behaviour exemplifies the complementary relationship between the generator and discriminator, emphasizing the DCGAN's effectiveness in generating high-quality synthetic images that enhance model training.

Overall, the findings suggest that leveraging DCGAN for synthetic data generation is a promising strategy in the field of plant disease recognition. Future research may expand upon this foundation by exploring alternative GAN architectures and further refining the generated images to optimize classification outcomes. The implications of this study extend beyond agricultural applications, potentially benefiting various domains where data scarcity hampers the performance of machine learning models.

Copyright:

References

[1] Moshou, Dimitrios, Michael A. S. M. Stamboulis, and Peter D. K. P. C. P. G. M. A. "Plant Disease Detection Based on Data Fusion of Hyper-spectral and Multi-spectral Fluorescence Imaging Using Kohonen Maps." *Real-Time Imaging* 11, no. 2 (2005): 75-83.

[2] Phadikar, Santanu, and Jaya Sil. "Rice Disease Identification Using Pattern Recognition Techniques." In *Proceedings of the 2008 11th International Conference on Computer and Information Technology*, 2008.

[3] Sanyal, P., and S. C. Patel. "Pattern Recognition Method to Detect Two Diseases in Rice Plants." *The Imaging Science Journal* 56, no. 6 (2008): 319-325.

[4] Camargo, A., and J. S. Smith. "An Image-Processing Based Algorithm to Automatically Identify Plant Disease Visual Symptoms." *Biosystems Engineering* 102, no. 1 (2009): 9-21.

[5] Bashir, Sabah, and Navdeep Sharma. "Remote Area Plant Disease Detection Using Image Processing." *IOSR Journal of Electronics and Communication Engineering* 2, no. 6 (2012): 31-34.

[6] Wang, Haiguang, X. H. J. L. H. L. S. M. C. Z. Y., and H. G. "Image Recognition of Plant Diseases Based on Backpropagation Networks." In *2012 5th International Congress on Image and Signal Processing*, 2012.

[7] Wang, Haiguang, and Wang Y. "Image Recognition of Plant Diseases Based on Principal Component Analysis and Neural Networks." In *2012 8th International Conference on Natural Computation*, 2012.

[8] Phadikar, Santanu, Jaya Sil, and Asit Kumar Das. "Classification of Rice Leaf Diseases Based on Morphological Changes." *International Journal of Information and Electronics Engineering* 2, no. 3 (2012): 460-463.

[9] Sanyal, P., and S. C. Patel. "Pattern recognition method to detect two diseases in rice plants." *The Imaging Science Journal* 56.6 (2008): 319-325.

[10] Camargo, A., and J. S. Smith. "Image Pattern Classification for the Identification of Disease-Causing Agents in Plants." *Computers and Electronics in Agriculture* 66, no. 2 (2009): 121-125.

[11] Al-Hiary, Heba, R. A. A. Al-Omari, O. H. K. A. H. H. J. O. M. A., and T. J. O. A. A. H. "Fast and Accurate Detection and Classification of Plant Diseases." *International Journal of Computer Applications* 17, no. 1 (2011): 31-38.

[12] Bashir, Sabah, and Navdeep Sharma. "Remote area plant disease detection using image processing." *IOSR Journal of Electronics and Communication Engineering* 2.6 (2012): 31-34.

[13] Neumann, Marion, et al. "Erosion Band Features for Cell Phone Image Based Plant Disease Classification." In *2014 22nd International Conference on Pattern Recognition*, 2014.

[14] Khirade, Sachin D., and A. B. Patil. "Plant Disease Detection Using Image Processing." In *2015 International Conference on Computing Communication Control and Automation*, 2015.

[15] Mohanty, Sharada P., David P. Hughes, and Marcel Salathé. "Using Deep Learning for Image-Based Plant Disease Detection." *Frontiers in Plant Science* 7 (2016): 1419.

[16] Sladojevic, Srdjan, et al. "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification." *Computational Intelligence and Neuroscience* (2016).

[17] Grinblat, Guillermo L., et al. "Deep Learning for Plant Identification Using Vein Morphological Patterns." *Computers and Electronics in Agriculture* 127 (2016): 418-424.

[18] Barbedo, Jayme Garcia Arnal, Luciano Vieira Koenigkan, and Thiago Teixeira Santos. "Identifying Multiple Plant Diseases Using Digital Image Processing." *Biosystems Engineering* 147 (2016): 104-116.

[19] Fuentes, Alvaro, et al. "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition." *Sensors* 17, no. 9 (2017): 2022.

[20] Sethy, Prabira Kumar. "Rice Leaf Disease Image Samples." Mendeley Data, V1 (2020). doi:10.17632/fwcj7stb8r.1.

[21] Laddha, Saloni, and Vijay Kumar. "DGCNN: Deep Convolutional Generative Adversarial Network Based Convolutional Neural Network for Diagnosis of COVID-19." *Multimedia Tools and Applications* (2022).

[22] Goodfellow, Ian, et al. "Generative Adversarial Nets." In *Advances in Neural Information Processing Systems*, 27 (2014).

[23] O'Shea, Keiron, and Ryan Nash. "An Introduction to Convolutional Neural Networks." *arXiv Preprint* (2015): arXiv:1511.08458.

[24] Lu, Yang, et al. "Identification of Rice Diseases Using Deep Convolutional Neural Networks." *Neurocomputing* 267 (2017): 378-384.