Chapter 5

CHAPTER 5

PROPOSED MODEL: A POSITIONAL-AWARE DUAL-ATTENTION AND TOPOLOGY-FUSION WITH EVOLUTIONARY GENERATIVE ADVERSARIAL NETWORK FOR HIGH-RESOLUTION DISEASED LEAF IMAGE GENERATION AND CLASSIFICATION

5.1 INTRODUCTION

This chapter presents the challenges of PDATFGAN and an exploitation of evolutionary algorithms in DL models. It also explores the application of evolutionary-based adversarial learning to enhance the creation and classification of high-resolution images of leaf diseases using a range of pre-trained DCNN models. against conclude, it compares the proposed Positional-aware Dual-Attention and Topology Fusion with Evolutionary Generative Adversarial Network (PDATFEGAN) against state-of-the-art models on the PVD dataset across a range of metrics.

5.2 CHALLENGES OF PGANs

Positional-aware GANs (PGANs) may be difficult because of the possibility of a non-convergent border set in the vicinity of equilibrium. This means that the generator and discriminator networks can reach a point where they are both producing images that are of high quality, but the images are not realistic. This is possible because the generator network may create unrealistically convincing visuals that can trick the discriminator network.

High-resolution leaf disease image creation with PGANs is complicated by the need for massive amounts of training data. To trick the discriminator network, the generator network must train itself to produce convincing fake images. However, due to the rarity of many leaf diseases, it might be challenging to acquire a substantial quantity of training data.

Finally, PGANs can be computationally expensive to train. This is due to the fact that training the generator and discriminator networks repeatedly is computationally costly. This can make it difficult to train PGANs on high-resolution images, as they

require a lot of computing power. By addressing the challenges of non-convergence, and computational expense, PGANs could be used to develop new diagnostic tools and treatments for leaf diseases.

5.3 REVIEW ON METHODS OF NON-CONVERGENCE PROBLEMS OF PGAN

Some of the research efforts that have been made to address the non-convergence problem of PGANs for high-resolution leaf disease image generation include:

- Using adversarial training with gradient penalty: This technique stabilizes the training process and stops the generator network from creating unrealistic images by adding a gradient penalty to the gradient of the discriminator network (Ororbia et al., 2017).
- Using a Wasserstein loss function: This loss function may assist to increase the produced pictures' realism the original data distribution is more closely reflected in the images generated by the generator network (Frogner et al., 2015).
- Using a hierarchical PGAN: This PGAN has two or more levels of generators and discriminators, which can help to improve the realism of the generated images at higher resolutions (Eghbal-zadeh et al. 2019)
- Using a conditional PGAN: This type of PGAN can be used to generate images that are conditioned on certain attributes, such as the class label. This can be helpful for generating more realistic images of specific leaf diseases (Dai et al. 2017)
- Using evolutionary PGAN: This PGAN model optimizes the creation of high-resolution photos of leaf diseases by combining PGAN and evolutionary algorithm concepts. Using evolutionary methods (Liu et al. 2022) that are motivated by natural selection, it seeks to increase the quality of produced content by enhancing the training and optimization process of PGANs.

5.3.1 Evolutionary Algorithms in GAN

Generative adversarial networks (GANs) are well-known for their data-generating prowess, and evolutionary methods are useful for both exploring many options and identifying the most effective set-ups. In the context of GANs, evolutionary algorithms

can be used to improve the generator by iteratively generating new images and selecting the ones that are most realistic (Bharti et al. 2022). These metaheuristic algorithms may be used to identify the best answer to a problem by searching through a pool of candidates in an iterative manner.

One way to use evolutionary algorithms in GANs is to combine them with a technique called adversarial training. Trainers and learners engage in a game-theoretic adversarial training session (Stinis et al., 2019). The generator works to produce synthetic pictures that are difficult to tell apart from actual ones, while the discriminator does the opposite. These techniques may be used to enhance the generator by feeding it a set of solutions that have a higher probability of being accurate.

GANs also make use of evolutionary algorithms in another capacity: designing the generator's underlying infrastructure. The input to the neural network used to create the picture is a vector of random noise. Evolutionary algorithms may be used to determine the optimal generator architecture, including the number of layers, the number of neurones per layer, and the activation functions (Lin et al. 2022).

Evolutionary algorithms have been shown to be effective in improving the quality of high-resolution leaf disease image generation. Some of the benefits of using these algorithms in GANs for high-resolution leaf disease image generation are

- By repeatedly exploring a population of solutions, evolutionary algorithms may be utilised to enhance the quality of the produced images
- These algorithms can be used to design the architecture of the generator, which can lead to better performance
- Evolutionary algorithms are relatively easy to implement and can be used with any GAN architecture

According to these significances, the evolutionary algorithm can be integrated with the PDATFGAN to search optimal generator structures for high-resolution leaf image creation and achieve faster convergence, compared to the other techniques discussed above.

5.3.2 Evolutionary Algorithm with Positional-aware GAN

Micro-scale coordinate system with fine granularity for G and a macro coordinate framework with a coarse granularity for D make up the PDATFGAN's generative network G, which uses a shallow-feature extraction network. Additionally, it takes into account whole pictures (original (a) and generated (x)), macro patches (original (a') and generated (x')), and micro patches (generated (x'')).

The primary objective of G is to generate realistic and faultless complete images by assembling a collection of x'' totally with a fusion factor φ and a cropping conversion ψ. In this case, a' is cut out of a using for ψ sampling the original macro patches in D. However, as shown in Fig. 5.1, the PDATFEGAN proposal develops an evolutionary algorithm that generates a population of generators (G) inside a specified discriminator (D). Each member of this population represents a possible answer inside a certain region of 's parameter space. During the evolutionary task, it is assumed that the populace makes climate changes on a frequent basis, which is evidence that the created generators can reliably provide highly realistic sample pictures and ultimately train the image distribution.

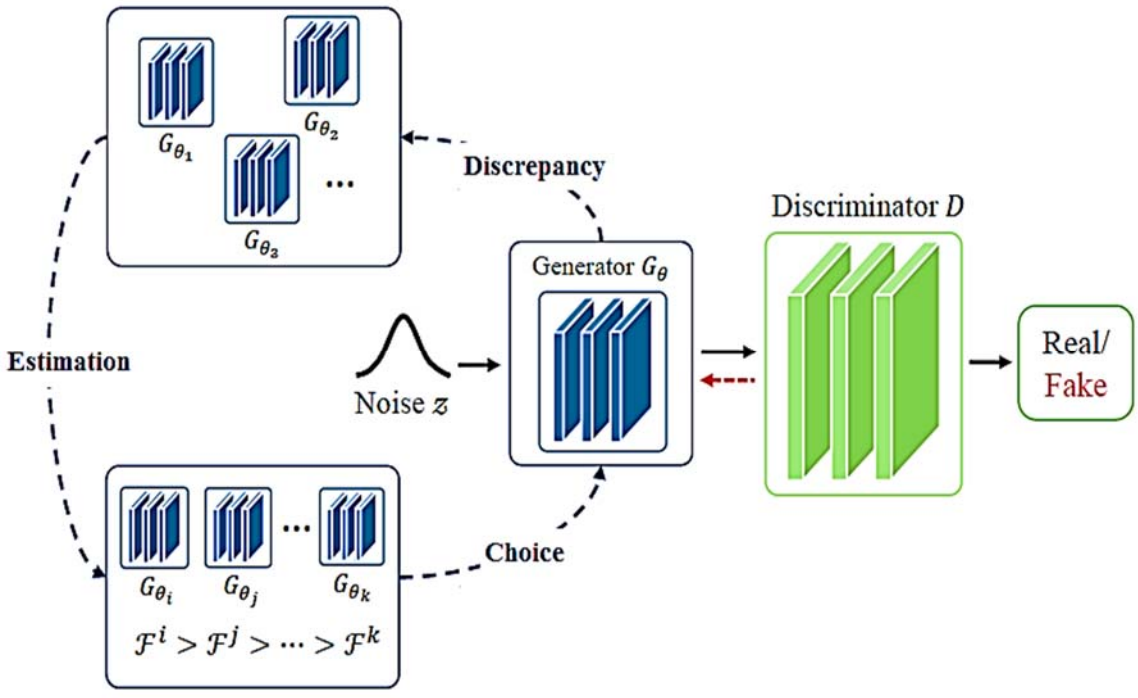

Fig. 5.1 Structure of Positional-aware Evolutionary GAN Model

There are three distinct stages at each evolutionary juncture:

- Discrepancy: The children $\{G_{\theta_1}, G_{\theta_2}, \ldots\}$ of a population member G_θ are generated using the discrepancy operators. To be more accurate, it is common practise to create many copies of a given individual or patent, each of which then undergoes its own unique mutations. After then, each modified copy is treated as a separate offspring.
- Estimation: The fitness factor $\mathcal{F}(\cdot)$ for each offspring is a function of the current D and provides an assessment of that offspring's efficiency or quality.
- Choice: Each offspring may be picked according to its fitness, and the least fit lineage can be wiped off. The remainder are preserved, and hence are able to reproduce and advance to the following generation.

At the end of each evolutionary stage, the environment D is tweaked such that the genuine macro patches a' can be distinguished from the fabricated ones x' produced by the refined generators. Furthermore, it helps G fool D with fake but realistic micro patches x'' made with the help of the produced generators,

$$L_{PEGAN} = \mathop{\mathbb{E}}_{a,c'}\left[log\, D\left(\psi(a, c')\right)\right] - \mathop{\mathbb{E}}_{z,C''}\left[log\left(1 - D\left(\varphi\left(G(z, C'')\right)\right)\right)\right] \qquad (5.1)$$

In Eq. (5.1), c' and C'' are coordinates for macro patches on D and micro patches on G and z is a latent vector. Thus, D may commonly provide the adaptive losses to motivate the G population to evolve toward superior solutions.

5.3.2.1 Mutation

Children are created by a combination of asexual regeneration and mutations in the process of evolution. Different learning goals, characterized by these mutation variables, purpose to reduce the discrepancies between the actual distribution and the distribution implied by the images. Consider that, in accordance with Eq. (1), the optimal discriminator $D^*(a) = \frac{a'}{a'+x''}$ is trained to assess the associated features of these mutations at each evolutionary stage.

Minimax Mutation

The minimax mutation characterizes the conventional GAN's minimax goal function, which is:

$$\mathcal{M}_G^{minimax} = \frac{1}{2}\mathbb{E}_{z,C''}\left[log\left(1 - D\left(\varphi\big(G(z,C'')\big)\right)\right)\right] \quad (5.2)$$

The goal of the minimax mutation, while searching for the best discriminator D^*, aims to reduce the Jensen-Shannon Divergence (JSD) between the new distribution and the original distribution of the image. Assuming the vanishing gradient and the support of two distributions on two manifolds, the JSD may be corrected. When $D\left(\varphi\big(G(z,C'')\big)\right) \to 0$, the gradient tends to zero since D is rejecting samples made with a high conviction.

If the constructed distribution overlaps with the image distribution, this suggests that D is unable to adequately distinguish actual from counterfeit samples, whereas the minimax mutation gives a strong gradient and frequently reduces the distance between the two.

Heuristic Mutation

As such, the goal of this heuristic mutation is to raise the log-probability that D is making a bad decision,

$$\mathcal{M}_G^{heuristic} = -\frac{1}{2}\mathbb{E}_{z,C''}\left[log\left(D\left(\varphi\big(G(z,C'')\big)\right)\right)\right] \quad (5.3)$$

While D throws away samples generated by the minimax mutation, the heuristic mutation does not reach saturation. Therefore, the heuristic mutation avoids the gradient vanishing problem and yields meaningful improvements in G. The reduction of the heuristic mutation is equivalent to the reduction of $[KL(x''\|a') - 2JSD(x''\|a')]$ for D^*. To be precise, KL inverted, -2 JSDs. When the JSD sign is negative, it means that the two distributions are drifting apart. In a purely virtual sense, this might lead to learning instability and variations in generator quality.

Least-Squares Mutation

A mutation known as least-squares employs least-squares targets to unjustly select for G in order to fool its D. Following is the least-squares mutation formula:

$$\mathcal{M}_G^{least-squares} = \mathbb{E}_{z,C''}\left[\left(D\left(\varphi\big(G(z,C'')\big)\right) - 1\right)^2\right] \quad (5.4)$$

The least-squares mutation is non-saturating whereas D can identify the produced sample i.e., $D\left(\varphi\left(G(z,C'')\right)\right) \to 0$. If D's result rises, then the least-squares mutation saturates and shift towards 0. D has a large advantage over G, and this will stop the gradient from going to zero.

5.3.2.2 Estimation

Estimation in this algorithm refers to the method of determining an individual's merit. The direction of evolution, or an individual's preference, is established by the development of an estimate or fitness factor that helps to evaluate the efficacy of mature people, or offspring. The quality and variety of the generated samples are prioritized by this algorithm. At first, 's output photos are fed into D, and the quality fitness score is measured as the mean of these results.

$$\mathcal{F}_q = \mathop{\mathbb{E}}_{z,C''}\left[\left(D\left(\varphi\left(G(z,C'')\right)\right)\right)\right] \tag{5.5}$$

To represent the quality of G throughout all evolutionary or adversarial stages, D is usually tuned to optimize learning. If G improves its quality score enough, it may be able to trick D with its fabricated samples, and the resulting distribution will be very close to that of the images. Additionally, efforts are made to improve the stability of learning by increasing the variety of the produced samples. The phrase of gradient-based regularization is used to characterize the diversity fitness score as:

$$\mathcal{F}_d = -log\left\|\nabla_D - \mathop{\mathbb{E}}_{\hat{x}'}[log\, D(\hat{x}')] - \mathop{\mathbb{E}}_{z,C''}\left[log\left(1 - D\left(\varphi\left(G(z,C'')\right)\right)\right)\right]\right\| \tag{5.6}$$

In Eq. (5.6), $\hat{x}' = \varepsilon x' + (1-\varepsilon)x'$ is calculated between randomly connected x' and a'using a random value $\varepsilon \in [0,1]$. The diversity of generated samples is calculated using the log gradient of varying D. The modified 's manufactured samples will disperse enough for preventing D to have obvious defenses if it achieves a relatively high diversity score, which correlates to tiny D gradients. This allows for a more consistent learning environment since D is able to update more smoothly. The estimate factor of this evolutionary method, which makes use of these two fitness measures, is given as:

$$\mathcal{F} = \mathcal{F}_q + \gamma\mathcal{F}_d \tag{5.7}$$

The optimal value of $\gamma \geq 0$ in Eq. (5.7) strikes a balance between generating quality and variety. Increased efficiency in both learning and generation likely to accompany a fitness score of $\mathcal{F}$ that is relatively high. Therefore, high-resolution images of plant leaves are collected with the PDATFEGAN.

5.4 BUILDING THE PROPOSED MODEL

The PDATFEGAN model is developed as a solution to the non-convergent boundary set near equilibrium problem of the generator in PDATFGAN model integrating evolutionary algorithm for high-resolution leaf disease image generation. The model has been built by combining the PGAN and evolutionary algorithms. The Plant Village Dataset (PVD) presented in Chapter 3 is used for the experiments. The stages engaged in the proposed PDATFEGAN are illustrated in Fig 5.2, wherein dataset preparation is given in Chapter 4. The second stage such as high-resolution leaf disease image generation is performed using the PDATFEGAN. A final stage such as the leaf disease classification using ShuffleNetV2, DenseNet121, and MobileNetV2 is presented in Chapter 4.

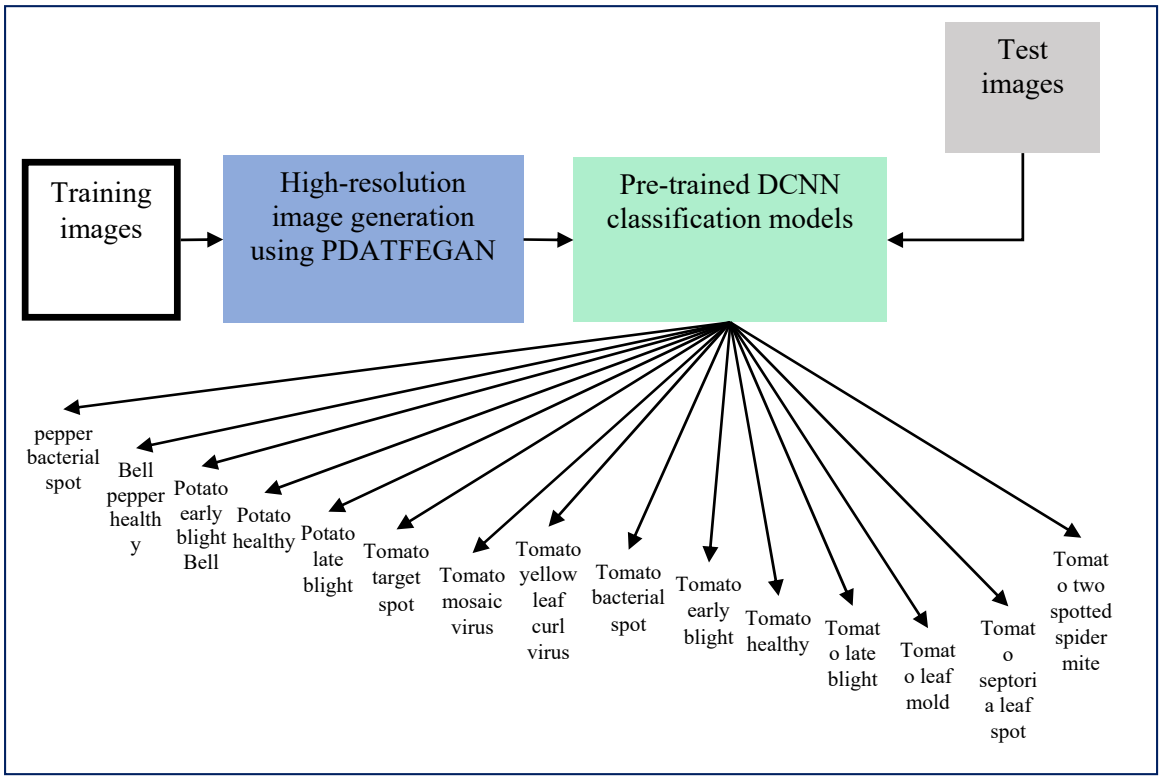

Fig. 5.2 Block Diagram of Proposed Model of PDATFEGAN

5.4.1 High-Resolution Image Generation using PDATFEGAN

After splitting training and test datasets, the PDATFEGAN model is performed as per the steps described on section 5.2.2, for generating high-resolution leaf disease images in the training set with faster convergence. In the PDATFEGAN, the network is built and trained using the parameters given in Table 5.1 with training images.

Table 5.1 Parameters for PDATFEGAN

Parameters	Range
Optimizer	Adam
Learning rate	0.0001
Number of epochs	180
Momentum	0.99
Weight decay	0.0001
Mini-batch size	64
Discriminator's updating steps per iteration	2
Number of parents	1
Number of mutations	3

As described in Section 4.4.3, the resulting high-resolution leaf disease pictures are next categorized into distinct disease classes using the various pre-trained DCNN classifiers.

5.5 RESULTS AND DISCUSSION

Here, researchers compare the proposed PDATFEGAN model to previously-existing models using a variety of performance assessment measures. Chapter 3 explains the datasets, assessment measures, and system settings in depth. Table 5.2 displays the values of precision, recall, f-measure, and accuracy for the ShuffleNetV2 classifier model on the PVD raw dataset, the PVD augmented by the DATFGAN, PDATFGAN, and PDATFEGAN models.

Table 5.2 Comparison of the proposed PDATFEGAN model using ShuffleNetV2

Performance Evaluation Metrics	Raw dataset	Dataset enhanced by DATFGAN	Dataset enhanced by PDATFGAN	Dataset enhanced by PDATFEGAN
Precision	0.8954	0.9135	0.9148	0.9232
Recall	0.8958	0.9140	0.9151	0.9177
F-measure	0.8957	0.9142	0.9150	0.9205
Accuracy	89.58%	91.38%	91.52%	92.36%

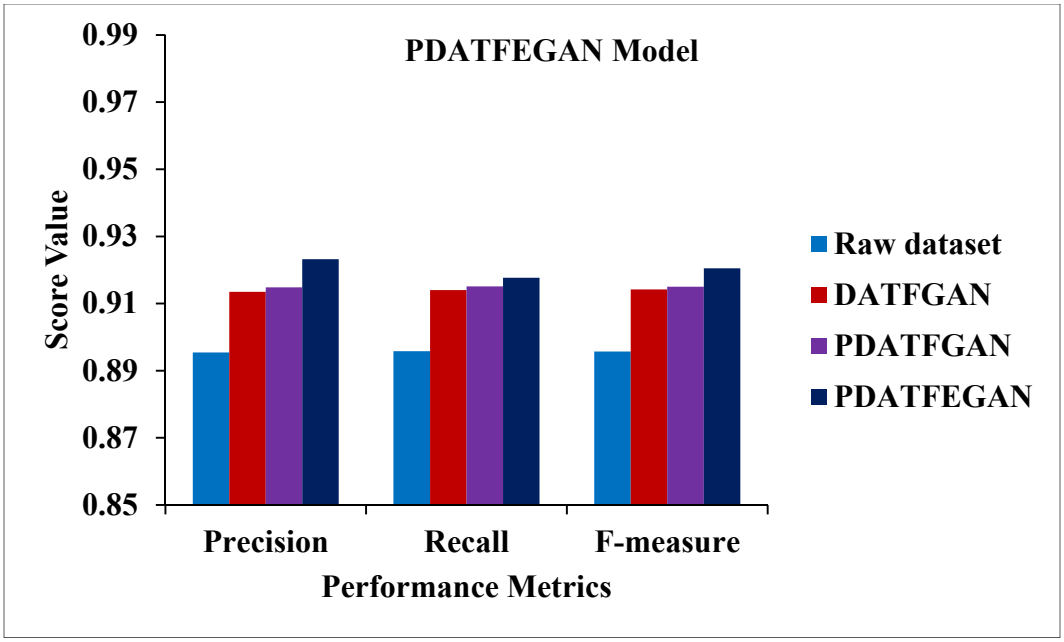

Fig. 5.3 Result of proposed PDATFEGAN Model Using ShuffleNetV2

Comparing precision, recall, and f-measure, the ShuffleNetV2 classifier performs better than both the raw and improved PVD models that use various GAN models. When pitted against the raw dataset, DATFGAN, and PDATFGAN models, PDATFEGAN-ShuffleNetV2 comes out on top in terms of precision, recall, and f-measure. It is observed that the precision increases by 3.1%, 1.06% and 0.92%, while the recall increases by 2.44%, 0.4% and 0.28% compared to the raw dataset, DATFGAN-ShuffleNetV2 and PDATFGAN-ShuffleNetV2, respectively. The f-measure also improves by 2.77%, 0.69% and 0.6% compared to the raw dataset, DATFGAN-ShuffleNetV2 and PDATFGAN-ShuffleNetV2, respectively, as illustrated in Fig 5.3.

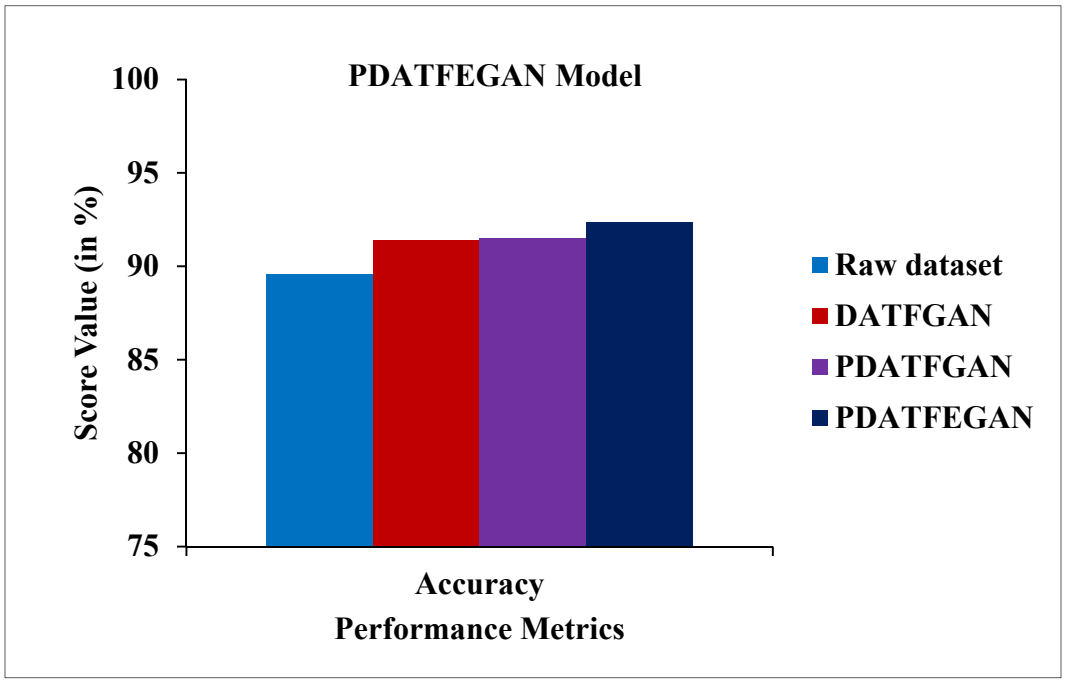

Fig. 5.4 Accuracy Comparison of PDATFEGAN Model Using ShuffleNetV2

A performance of the ShuffleNetV2 classifier tested using raw PVD and enhanced PVD using different GAN models is depicted in terms of accuracy. It is shown that the accuracy of PDATFEGAN-ShuffleNetV2 is improved up to 3.1%, 1.1% and 0.9% compared to the ShuffleNetV2 using the raw dataset, DATFGAN-ShuffleNetV2 and PDATFGAN-ShuffleNetV2, respectively. This is achieved by optimizing the high-resolution leaf image generation processing using PEGAN as shown in Fig 5.4.

Table 5.3 displays the results of the DenseNet121 classifier's tests on the PVD raw dataset, the PVD augmented by the DATFGAN, PDATFGAN, and PDATFEGAN models, and the DenseNet121 classifier's tests on the PDATFEGAN dataset.

Table 5.3 Comparison of the proposed PDATFEGAN model using DenseNet121

Performance Evaluation Metrics	Raw dataset	Dataset enhanced by DATFGAN	Dataset enhanced by PDATFGAN	Dataset enhanced by PDATFEGAN
Precision	0.8841	0.9249	0.9270	0.9321
Recall	0.8843	0.9252	0.9273	0.9331
F-measure	0.8842	0.9251	0.9272	0.9326
Accuracy	88.47%	92.54%	92.74%	93.26%

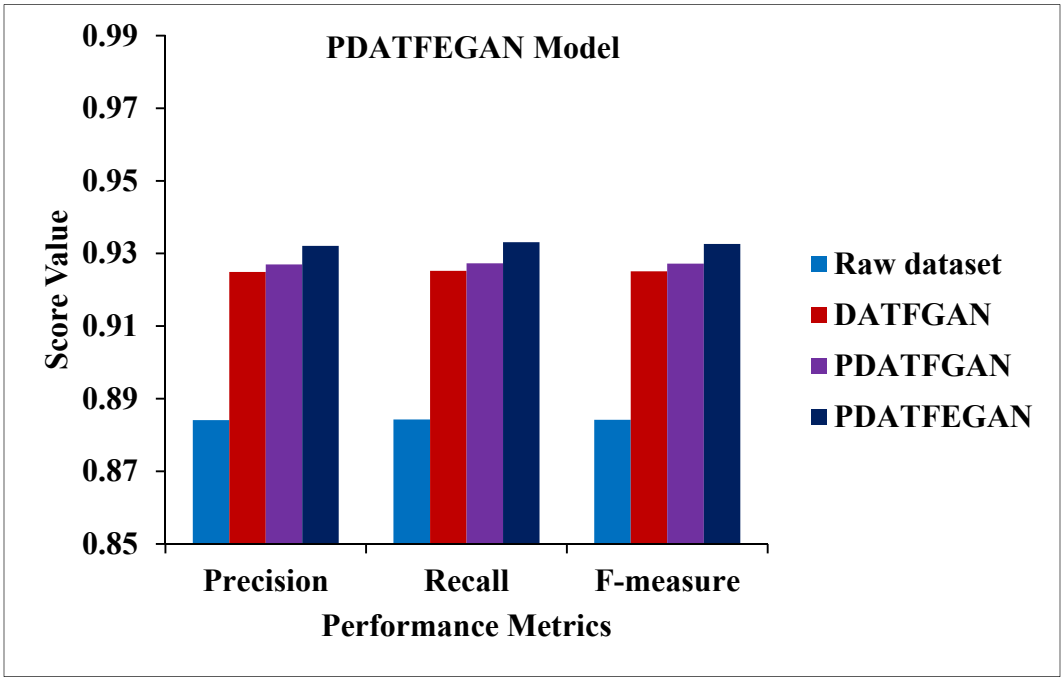

Fig. 5.5 Result of Proposed PDATFEGAN Model Using DenseNet121

The DenseNet121 outperforms the raw and refined PVD models trained using several GAN models in terms of precision, recall, and f-measure. The PDATFEGAN-DenseNet121 obtains improved precision, recall and f-measure compared to the raw

dataset, DATFGAN and PDATFGAN models as seen in Fig 5.5. Compared to the raw dataset, DATFGAN-DenseNet121 and PDATFGAN-DenseNet121 are shown to improve accuracy by 5.43%, recall by 0.78%, and F1 score by 0.55%. The f-measure also improves by 5.47%, 0.81% and 0.58% compared to the raw dataset, DATFGAN-ShuffleNetV2 and PDATFGAN-ShuffleNetV2, respectively.

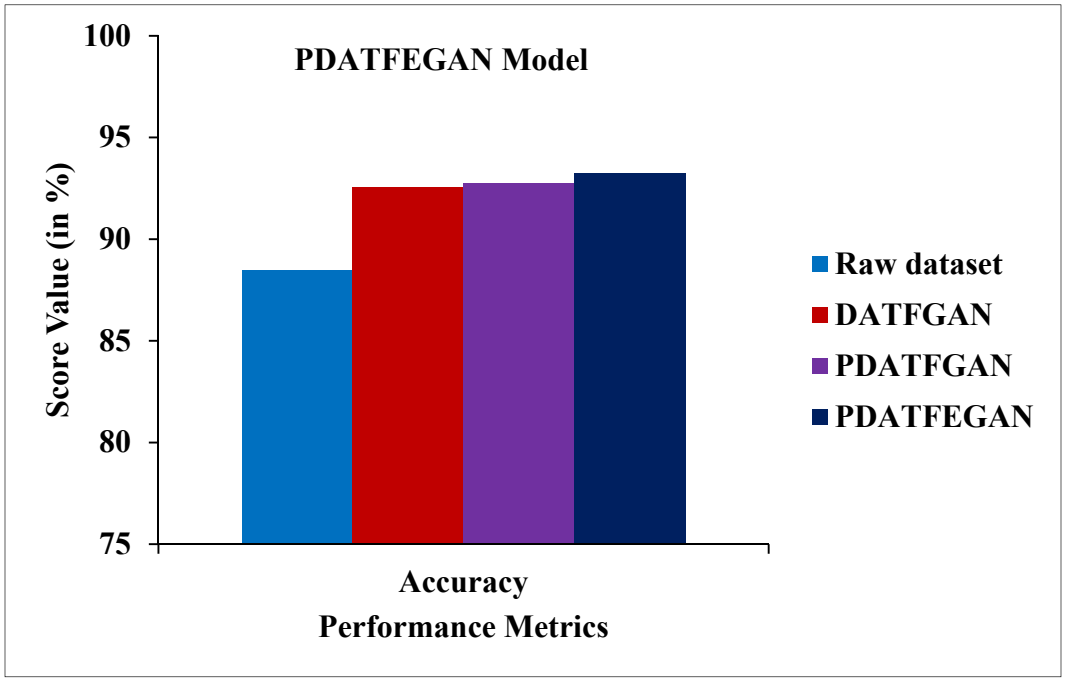

Fig. 5.6 Accuracy Comparison of PDATFEGAN Model Using DenseNet121

DenseNet121 classifier's accuracy when evaluated with raw PVD and improved PVD utilizing various GAN models is shown. It is observed that the accuracy of PDATFEGAN-DenseNet121 is raised to 5.41%, 0.78% and 0.56% than the DenseNet121 using the raw dataset, DATFGAN-DenseNet121 and PDATFGAN-DenseNet121, respectively. This is achieved by optimizing the PGAN for high-resolution leaf image generation based on the evolutionary algorithms, as shown in Fig 5.6.

Table 5.4 displays the results of tests conducted on the MobileNetV2 classifier model using the PVD raw dataset, upgraded PVD by the DATFGAN, PDATFGAN, and PDATFEGAN models.

Table 5.4 Comparison of the proposed PDATFEGAN model using MobileNetV2

Performance Evaluation Metrics	Raw dataset	Dataset enhanced by DATFGAN	Dataset enhanced by PDATFGAN	Dataset enhanced by PDATFEGAN
Precision	0.9062	0.9264	0.9283	0.9351
Recall	0.9065	0.9266	0.9287	0.9362
F-measure	0.9064	0.9265	0.9285	0.9356
Accuracy	90.66%	92.69%	92.87%	93.58%

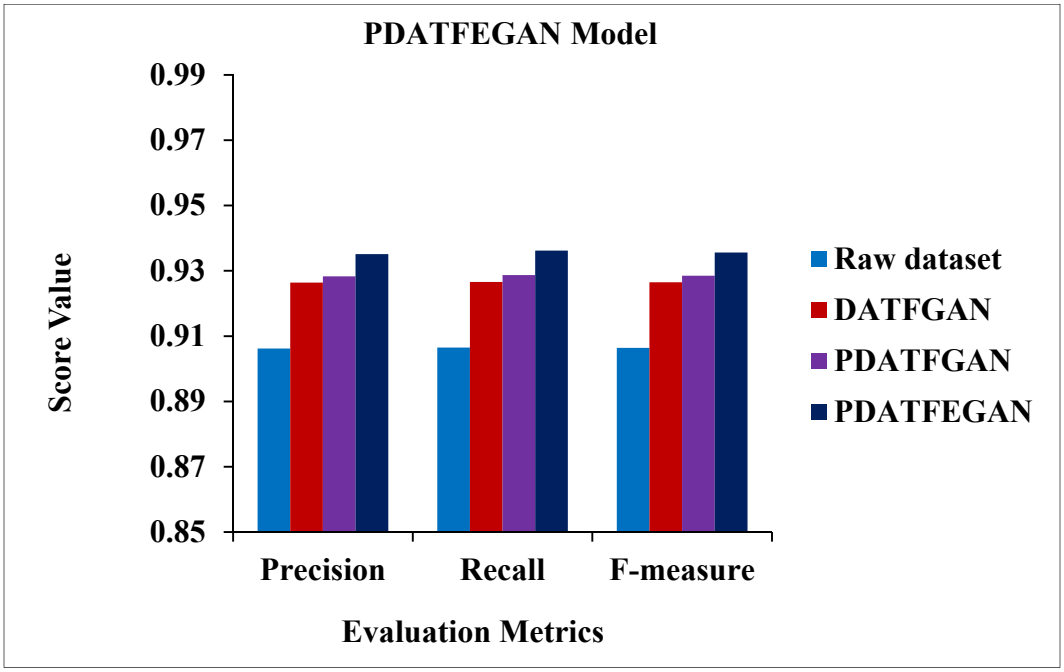

Fig. 5.7 Result of Proposed PDATFEGAN Model Using MobileNetV2

When compared to both the raw and improved PVD models that make use of other GAN models, MobileNetV2 performs better across the board. Figure 5.7 shows that when compared to the raw dataset, the DATFGAN model, and the PDATFGAN model,

the PDATFEGAN-MobileNetV2 obtains better results across the board. When comparing the raw dataset to DATFGAN-MobileNetV2 and PDATFGAN-MobileNetV2, it is shown that the precision rises by 3.19%, 0.94%, and 0.73%, while the recall increases by 3.28%, 1.04%, and 0.81%. The f-measure also improves by 3.22%, 0.98% and 0.76% compared to the raw dataset, DATFGAN-ShuffleNetV2 and PDATFGAN-ShuffleNetV2, respectively.

A performance of the MobileNetV2 classifier tested using raw PVD and enhanced PVD is depicted in terms of accuracy. It is noted that the accuracy of PDATFEGAN-MobileNetV2 is improved by 3.22%, 0.96% and 0.76% compared to the MobileNetV2 using the raw dataset, DATFGAN-MobileNetV2 and PDATFGAN-MobileNetV2, respectively as shown in Fig 5.8. This is accomplished by adopting evolutionary algorithm for optimizing the generator function to create high-resolution leaf images efficiently.

The results of these comparisons show that the MobileNetV2 classifier outperforms the other classifiers on both raw and enhanced PVD. The MobileNetV2 with the PDATFEGAN model achieves higher accuracy when classifying leaf diseases.

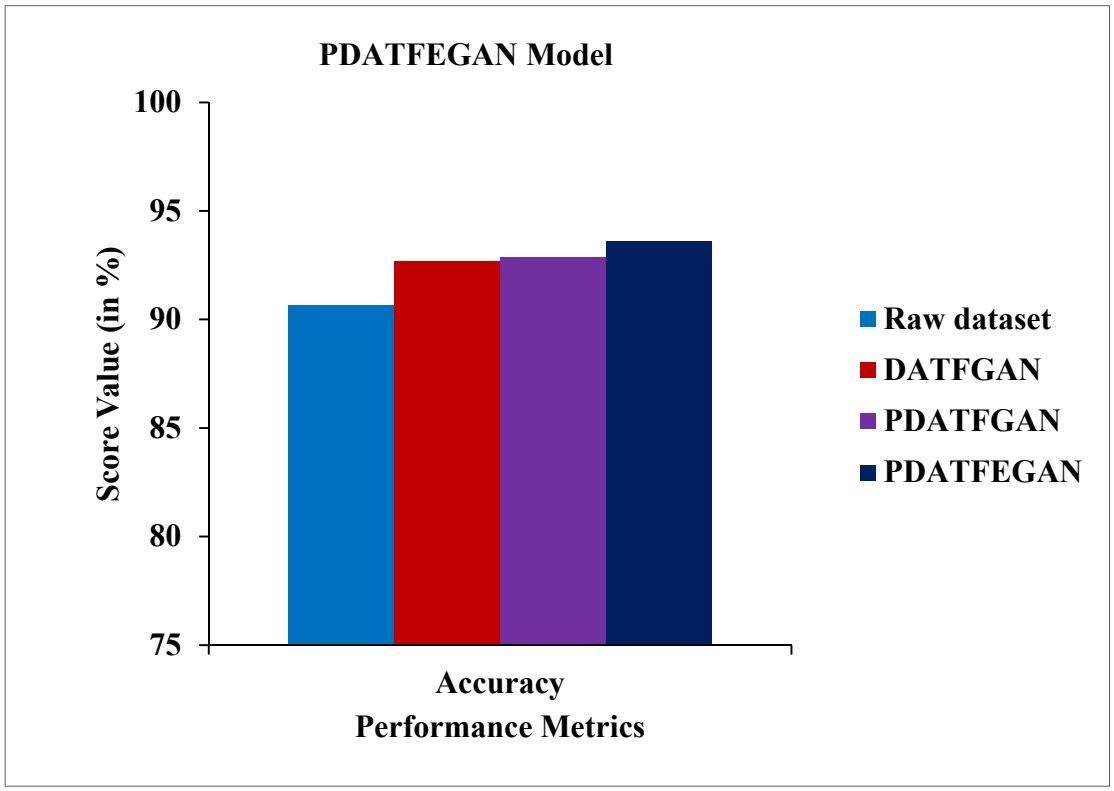

Fig. 5.8 Accuracy Comparison of PDATFEGAN Model Using MobileNetV2

5.6 SUMMARY

To summarize, an overview of evolutionary algorithms used in GAN and their benefits in high-resolution leaf disease image generation is presented in this chapter. The design and development of the new PDATFEGAN model combining PGAN and evolutionary algorithm is described. The models are evaluated on a wide range of leaf disease image types. The findings show that the suggested PDATFEGAN model is useful for image classification of leaf diseases.