Synthetic Data Augmentation of Tomato Plant Leaf using Meta Intelligent Generative Adversarial Network: Milgan

Sri Silpa Padmanabhuni, Pradeepini Gera
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

***Abstract*—Agriculture is one of the most famous case studies in deep learning. Most researchers want to detect different diseases at the early stages of cultivation to save the farmer's economy. The deep learning technique needs more data to develop an accurate system. Researchers generated more synthetic data using basic image operations in traditional approaches, but these approaches are more complicated and expensive. In deep learning and computer vision, the system's accuracy is the crucial component for deciding the system's efficiency. The model's precision is based on the image's size and quality. Getting many images from the real-world environment in medicine and agriculture is difficult. The image augmentation technique helps the system generate more images that can replicate the physical circumstances by performing various operations. It also prevents overfitting, especially when the system has fewer images than required. Few researchers experimented using CNN and simple Generative Adversarial Network (GAN), but these approaches create images with more noise. The proposed research aims to develop more data using a Meta approach. The images are processed using kernel filters. Different geometric transformations are passed as input to the enhanced GANs to reduce the noise and create more fake images using latent points, acting as weights in the neural networks. The proposed system uses random sampling techniques, passes a few processed images to the generator component of GAN, and the system uses a discriminator component to classify the synthetic data created by the Meta-Learning Approach.**

***Keywords*—Basic image operations; meta-learning techniques; generator; discriminator; synthetic data; sampling techniques; latent points; kernel filters**

I. Introduction

Deep Learning algorithms are famous for solving case studies related to medicine and agriculture. The efficiency of the deep learning model depends on the selection of neural network design. The process of defining the best estimators for a network is coined "Hyper Turning." he system needs more balanced and huge of data to estimate the proper parameters that suit the network. It is highly impossible to collect the real-time images from the farming lands using digital cameras, and the collection of satellite images covers the entire crop. However, it cannot provide individual leaf analysis [19]. So, the agriculture industry needs a system that can generate similar images covering controlled and uncontrolled situations like blur leaf images due to heavy wind effects, half leaf images due to the distance, and others. The generation of similar images is known as "Data Augmentation," and the data is known as “Synthetic Data.” The previous researchers created synthetic data using basic image manipulation techniques, GANs or Auto encoders and decoders [10]. The number of images will increase using manipulation techniques like transformation and rotation, but they cannot produce uncontrolled images. Researchers have introduced GANs and encoders over the past few years to create compressed and noisy images so they can handle uncontrolled conditions. However, due to single components of GANs, they produce more noise than required. The proposed system to reduce the number of noisy images and enhance the quality of controlled images introduced a meta-intelligent environment where the architecture is improved by increasing the components of GAN, and these components take the manipulated images rather than the original images from the GAN. The proposed research generates images using simple operations, as discussed below.

A. Basic Image Operations

Traditionally, to manipulate the images, researchers performed Geometric transformations [18], which involve rotations, flipping, resizing, shifting, zooming, cropping, and noise operations. The system can also perform color space transformations. Images are composed of RGB color conventions but must be converted into grey-scale or other color saturation values for efficient data processing. The essential operations involve a kernel filter [16], extracting the region of interest from the image using convolution neural networks. The image is saved as a two-dimensional matrix, and the filter does a dot product between the inputs and filter layers and then adds the values to get a single value in each location. Edge detection, image sharpening, and blurring operations benefit from kernel filtering. The most recent improvements are the random erasing technique to remove pixels in an image by choosing a rectangular region based on the likelihood determined by aspect and area ratios. The Random Erasing technique is essential for image recognition, object detection, and people re-identification. It can be used on pre-trained models and easily combined with neural networks [13]. The different types of image operations are shown in Fig. 1.

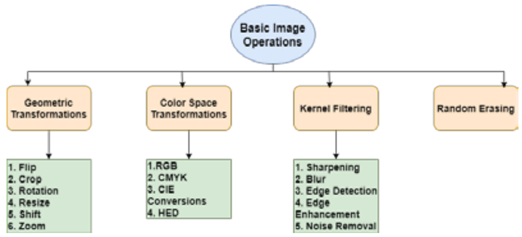

Fig. 1. Types of basic Image Operations.

1) Rotation: Rotation is the most straightforward operation that can be applied to any image to change its orientation. The rotation angle is expressed in degrees, ranging from 0 to 360 clockwise[21].

2) Flipping: It is a variation in the rotational operation. The transposition of row and column pixels is accomplished using a flipping operation. Two types of flipping operations can be carried out here. Horizontal and vertical flips may be used, depending on the quality of the images.

3) Shearing: Shearing is moving pixels from one location to another, either horizontally or vertically. The image's dimensions remain unchanged. Only a few pixels will be clipped off.

4) Cropping: The images generally have a region of interest at different locations. Cropping the image at the required area is the best operation to identify the location. Since the proposed system is a classification model, the model's output should be a subset of the entire image[17].

5) Zooming: When a system zooms in, the system can either add new pixel values or interpolate existing ones. If we take the value x as an example, the zoom will perform 1+x and 1-x operations on the image pixels. It aids in the image's standardized sampling.

6) Brightness or contrast: This operation can darken or brighten an image. This operation primarily assists the model in training the device in various conditions. The system may use percentages to specify the minimum and maximum values. The image is darkened if the value is less than 1.0 and brightened if the value is more than 1.0.

The below section presents the algorithm in which all the manipulation operations are randomly applied to the dataset's images.

Algorithm: Basic Image Operations

Input: Load the tomato dataset from the data repository, T_Data

Output: Retransformed Images

1. for i in len(T_Data):

Resize_timage←ImageGenerator(T_Data,rescale=1/255).

2. train_img_gen←ImageDataGenerator(rotation_range=45, width_shift_range=0.20, height_shift_range=0.25, horizontal_flip= True, zoom_range=0.50)

3. img_gen←train_img_gen.flow_from_directory(directory=path,batch_size=32)

4. for img in img_gen:

imshow(img)

The algorithm will give the individual operations augmented images, as shown in Fig. 2(a), and the combined basic image operations are shown in Fig. 2(b).

Fig. 2(a) represents the individual operation on each image, whereas the proposed system uses the "Data Generator" module to apply different processes on the dataset images. Fig. 2(b) represents five such figures as sample output.

B. Data Augmentation

Manually increasing the size of the image dataset is complicated. Neural networks provide different "data augmentation through GAN" [11], as shown in Fig. 3, to simplify the process. The data augmentation can be performed in either online mode or offline mode. This research implements online augmentation techniques since the offline [12] mode involves high memory usage, which is expensive and time-consuming. In online data augmentation, images are transformed randomly in different batches, and the model is trained with more cases in each epoch.

The two design considerations for augmentations are Train-time data augmentation to minimize generalization error and test-time data augmentation to increase predictive accuracy. The most critical operation to perform is resizing. Resizing ensures that all images are the same size. Specific pixel scaling and normalization operations can be carried out.

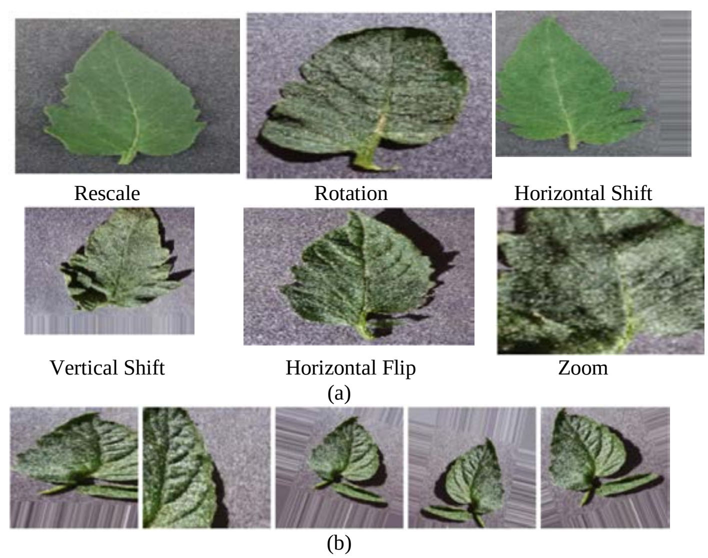

Fig. 2. (a) Individual basic Image Operations on Tomato Leaves, (b) Combined basic Image Operations using Image Data Generator.

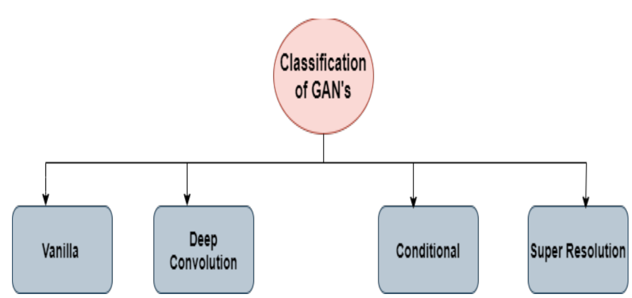

Fig. 3. GAN's Classification.

GAN technique provides adversarial preparation in critical models where Gaussian noise is injected, and images are transformed with worst-case perturbations [15], resulting in incorrect results with high confidence values. It also includes a neural style transferring technique that aids in the creation of new images by blending them, mainly when the model contains text or kinds. It employs two distinct distance roles, one for identifying text differences and the other for identifying style differences. In GANs, the generator takes a fixed-length vector as input and gives a domain-specific sample as output. Discriminator [14] takes the domain's information and generates a binary value depending on the prediction.

The entire paper organizes into five sections, where the introduction discusses the techniques related to image manipulation and data creation operations. The literature review section focuses on the previous researchers' methodologies and their merits and demerits. The literature section and also represents the gaps in the earlier works by analyzing the fallbacks associated with each method. The proposed process initially defines the architectural difference between the existing and proposed techniques. It also presents the layered architecture of the discriminator component and generator component. Finally, it describes the overall process of creating the synthetic data to enhance the size of the data. The result section describes how the accuracy of creating fake and real images in every epoch is computed. It also proves state-of-the-art compared with the other works mentioned in the literature survey. The paper in the last section concludes the piece by justifying the need for intelligent approaches to enhancing data size and presents the future to design an automated system for agriculture [21-24].

II. Literature Review

A. Methodologies Implemented using Researchers

[1] Ahmed Ali et al. proposed a framework in which GANs are integrated with CNN to classify mosaic virus in tomato plants. The author increased the dataset size by generating fake images stage by stage. This research's stages are healthy, early, and late infection. Random noise and each stage are passed as discriminators' inputs to generate synthetic samples. In hidden layers, global pooling is applied to create fake images almost nearer to the real images in the training dataset. It generates 512 feature vector dimensions and is passed as input to the discriminator to check its authenticity. If the accuracy rate is low, it loops the entire process for that particular image until it obtains a high accuracy rate. Finally, the model takes these augmented images as input to the neural network and performs multi-classification. CNN uses a 4-layered architecture by taking the standard activation function "ReLu" and standard optimizer "ADAM." This model balanced the dataset by producing an equal number of images in each stage.

In [2], Umit Atila et al. experimented with EfficientNet to eliminate the problems caused by the ImageNet pre-trained Model. The first step to improving efficiency is to perform the uniform scaling operation in all directions along with uniform translations and rotations. The uniform scaling factor is measured by drawing the grid relation between neighboring pixels. The obtained values act as the constraints based on which a neural network is designed to predict the class labels approximately. This research constructed inverted bottleneck layers available in Mobilev2Net to generate the images based on the expansion and contraction techniques. Transfer learning replaces the last layers of the B5 model, with 16 layers of MobileNet and four fully connected layers as a classifier. The disadvantage of this model is that it initialized the learning rate as 0.0001 but obtained an accuracy of 99.97%, which means this model cannot learn the complex relationships among the features.

In [3], Hongxia Deng et al. designed RHAC_GAN using the hidden variable mechanism to increase the dataset size because the collection of small data would not make many effects and would not get better results by using the CNN algorithm. Traditional approaches used image rotation, increased image brightness, and many other factors to solve this problem. Even these cannot help collect extensive data, which results in less accuracy. In the case of agriculture, it needs a large amount of data to identify the disease; however, due to insufficient data collected, even the best method, CNN, cannot give the best results. For the more extensive data collection, they proposed an ACGAN method that takes the generated images and divides them into parts to visualize the disease better. The hidden parameters in this model aim at low-frequency parameters in the image and show it as the disease-affected place.

In [4], Yang Wu et al. implemented an Adversarial VAE encoding mechanism for increasing the number of tomato leaf images in the dataset. Traditional VAE is extended by replacing the single scaling component with a multi-residual member for generating different scaling images in different directions. It is not easy to extract and train the parameters in the model, so the system needs a generalized learning environment and the labels associated with images. In this Model, GAN is integrated with the encoder in stage 1 to create compressed images as a dummy dataset. In stage 2 decoder is attached to reconstruct by creating latent space variables. Usage of the mean pre-processing technique helps the model enhance the dimension from 128 to 256 during the encoding process. The downsampling of neural networks helps the model create a scale block using reduced mean operation.

In [5], Amreen Abbas et al. designed conditional GAN using a transfer learning technique on nine diseases. The model uses a generator of CGAN to create augmented images using three layers. One of the three layers is implemented using an embedded pairing mechanism, consisting of a series of flatten and dropout layers. The advantage of CGAN lies in the fast creation of enhanced images by learning approximately seven lakhs of training parameters, and all the images are labeled from 0 to 9. In the next phase, the augmented images are merged with DenseNet using max pooling and average pooling layer alternately to minimize the number of features in every layer. A fine-tune layer is attached to the pre-trained network, which helps adjust weights optimally and perform the hyper tuning.

In [6], Jashraj Karnik et al. used a new way of classification called YOLOv3. KNN, K-means, and SVM did the data classification in the olden days. However, with the help of YOLOv3, it would be better and more accessible. They had proposed four ways for the farmers to understand the problem. Pre-processing, data augmentation, classification, and intersection over union have been done. The farmers who lived far away would face a problem identifying the disease.

Moreover, with the help of deep learning and AI, they can only identify the disease in the early stage. All farmers who do not know which plant to have or not and what fertilizers to use were also present in the AI, which would help the farmers increase their growth. YOLOv3 would work in 2 ways: classifying the leaves and the disease. This way, the output would be given to the CNN, identifying the disease. In their proposed AI, they would recognize the disease and help the farmers take the next step that would help them increase their growth.

In [7], Aaditya Prasad et al had implemented a two-step machine learning model for better accuracy for low- and high-fidelity pictures. Here they used UAV images. The two data collectors were used to reduce the imbalance in the high fidelity. One was a data generator, and the other was modeling, and each of these elements had two-step for identification and classification. This model identifies the affected region and captures high-definition images for better understanding and classification. The second phase deals with the technical method for proletarian revolution: synthetic crop images in data generators.

In [8], using MATLAB, Bhattacharya developed a method based on CNN. The rapid investigation and diagnosis of plant illnesses can help manage disease development on various crops, enhancing harvest growth and yield. Apply image analysis and deep learning approaches to automatically discover acceptable features for differentiating the different types of plant disease to make the system efficient. The author introduced the method called convolution neural network architecture. This deep learning technique autonomously categorized three types of wheat leaf disease detection (fungal blight, blasting, and black mark) inside this study [25-28]. They have created a technique for recognizing normal and diseased plants from a batch of 1000 rice plants during the classification stage.

In [9], Guo et al. proposed a method based on RPN integrated with image Segmentation. This work provides a deep learning-based statistical equation for plant disease diagnosis and identification that increases the accuracy, flexibility, and training effectiveness. Its region proposal network is used to identify and locate the affected parts in the plants. So therefore, Chan–Vese (CV) method segments the image to find the illness attribute. Lastly, the separated leaves were fed into a transferring learning model, developed on a dataset containing diseased leaves inside a simplified environment. This prototype is also tested for brown rot, fungal disease, and rust disease. This analysis shows an accuracy of 83.57 percent, which is greater than the best way to minimize the disease impact on food production and promote agriculture's long-term sustainability. As a result, the deep learning methodology suggested in the research has a lot of potential in smart agriculture, environmental regulation, and agricultural output.

B. Comparison Analysis

Table I coins all the methodologies presented in the above section with their advantages and limitations to identify the gaps associated with each methodology to solve the problems efficiently.

TABLE I. COMPARATIVE ANALYSIS OF EXISTING APPROACHES

S.No	Author Name	Algorithms Used	Merits	Demerits
1	Ahmed Ali	GANs integrated with CNN	Since the model produces images stage by stage, it can productively increase the size of the dataset.	To increase the accuracy rate for each image, it iterates in loops, which makes the system expensive.
2	Umit Atila	EfficientNet	Because of the inverted layer architecture, more similar images are generated at a low cost by reducing the generation cycles from k*k to k.	It has hardware limitations because of which it can perform only binary classification. So, plants with multiple diseases cannot be predicted using this model.
3	Hongxia Deng	RAHC-GAN	Used for a large amount of data.	Have to explore more data to maintain high value and accuracy
4	Yang Wu	AVE	The architecture with complete dense layers helps the model to train the system with	The model requires a large amount of annotation data to identify the correct labels for each image
5	Amreen Abbas	CGAN using transfer learning	Usage of fine-tuned layers in DenseNet improves the multi-classification rate without overfitting.	The model needs retraining to work with compressed images efficiently.
6	Karnik	CNN	Here a pixel identification has been made	Rotational image identification is complex.
7	Aaditya Prasad	GAN	Had two steps approach for classification and identification	Had to increase the accuracy and classification model
8	Bhattacharya	CNN using MATLAB	Image-based classification has been done	Pre-trained the model based on size.
9	Guo	RPN and Image Segmentation, SVM	Here migration learning model is used to identify the diseased leaf.	Iteration calculation is significantly less when compared to other algorithms.

C. Research Gaps Identified and Solutions to Solve them

1) Most existing systems cannot produce good images using the basic operations because they are costly and cannot simulate real-world conditions.

2) Usage of CNN will complicate the model because the direct conversion of the image into a matrix will increase the training time.

3) Traditional GANs use the MNIST dataset to regenerate the images. We need to use 2-layered architecture to extract simple 28*28 images, which produce noisy images as fake images.

So, in the proposed research, instead of 3-layered architecture, the model presents a 4-layered Enhanced Cyclic GAN to extract the features from the image and reconstruct it as early as possible. To overcome the problem of the noisy image, the model initially performs pre-processing using kernel filters, and a few images are constructed using the basic image manipulations. These pre-processed and manipulated images are passed as input to Enhanced Cyclic GAN to create the synthetic data. This process is known as the "Meta-Learning Technique," as shown in Fig. 4.

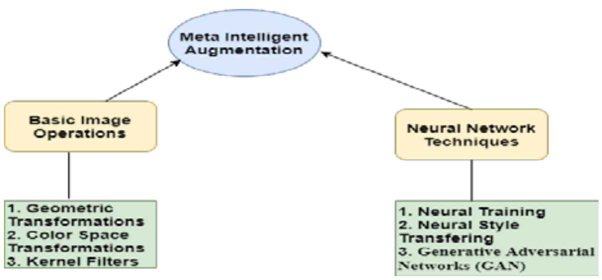

Fig. 4. Operations in Meta Intelligent Image Augmentation Process.

III. Proposed Methodology

Using machine learning and deep learning techniques, several Research Scholars and Scientists are constantly working on various plant diseases to improve production quality. These researchers used the PlantVillage dataset [20] to obtain plant images under controlled conditions. The use of computer vision and deep learning techniques have resulted in high-accuracy systems for detecting plant diseases automatically. The images of the dataset are presented in Fig. 5. Image augmentation creates new images in different scenarios using the existing images and applying popular deep learning techniques. The proposed paper discusses the various image augmentation techniques applied to the tomato plant's dataset from the PlantVillage open source. It also suggests the best method to use the data augmentation by comparing the performance evaluation metrics.

A. Pre-Processing

The best solution for solving the random and uneven noise is to apply filters to the images. Analyzing the dataset makes it clear that most of the images have salt and pepper noise. The proposed research used a median filter to reduce these noises and obtained the results shown in Fig. 6(b).

In Fig. 6(a), the original image contains more disturbances due to signal fluctuations. The median filter smoothens the images by replacing each pixel with the median of neighboring pixels. The main advantage of the median filter lies in its edge preservation and removing the spikes.

B. Discriminator

GAN works as a classifier to distinguish real and fake images. Real images are positive points, and counterfeit images are negative points. The discriminator gets input from training samples and the generator component of the GAN. The working of this component is based on conditional probability; this component calculates the chance for an image authenticity. The discriminator has associated loss functions because the system has accuracy in classifiers. The GAN structure imposes a penalty for misclassification. This component reduces the penalty by adjusting the latent points through the backpropagation technique. The primary focus of the loss function is to analyze the difference between sample distribution among real and fake images. The proposed research designed the layers in the discriminator component, as shown in Fig. 7.

C. Generator

It acts as a creative tool to generate fake images to mislead the discriminator in the classification process by maximizing the penalty and taking the feedback from the above component. Creating a random sample image induces noise as input and then converts it into necessary output. In the GAN structure, the training of the generator is the most crucial part, so the noise is converted into vector space. This space should be balanced by reducing one component and increasing another component's performance. During the training process, using the cross-entropy mechanism, GANs use the min-max loss function as shown in equation (1).

	
Target Spot	Mosaic Virus
	
Early Blight	Healthy

Fig. 5. Original Dataset Images of Different Diseases of Tomato.

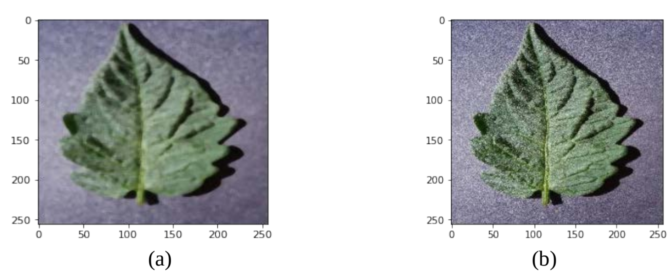

Fig. 6. (a): Original Image, (b): Filtered Image.

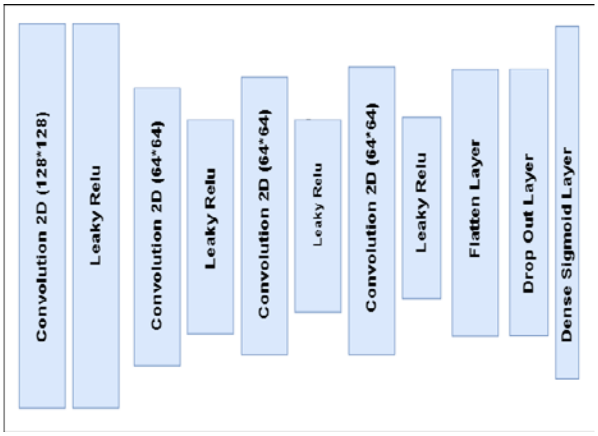

Fig. 7. Layers Design in Discriminator Module.

$$MinMax\_CGAN\_Loss = Expected\_Real(\log(Discriminator(X)) + Expected\_Fake(\log(1 - Discriminator(Generator(Y)))\text{- (1)}$$

Where

Discriminator(X) analyzes the probability associated with the real images.

Generator(Y) analyzes the probability associated with the fake images along with the noise.

The summary evaluation of the layered architecture of the generator component is represented in Fig. 8.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 65536)             6619136

 leaky_re_lu_4 (LeakyReLU)   (None, 65536)             0

 reshape (Reshape)           (None, 16, 16, 256)       0

 conv2d_transpose (Conv2DTra  (None, 32, 32, 128)      524416
 nspose)

 leaky_re_lu_5 (LeakyReLU)   (None, 32, 32, 128)       0

 conv2d_transpose_1 (Conv2DT  (None, 64, 64, 128)      262272
 ranspose)

 leaky_re_lu_6 (LeakyReLU)   (None, 64, 64, 128)       0

 conv2d_transpose_2 (Conv2DT  (None, 128, 128, 128)    262272
 ranspose)

 leaky_re_lu_7 (LeakyReLU)   (None, 128, 128, 128)     0

 conv2d_4 (Conv2D)           (None, 128, 128, 3)       18819

=================================================================
Total params: 7,686,915
Trainable params: 7,686,915
Non-trainable params: 0
```

Fig. 8. Summary Evaluation Report of Generator Component.

D. Enhanced Cyclic Generative Adversarial Networks

In practice, GANs act as a tool to generate a fake image by analyzing the sample distribution of the real image. GANs should be designed with the help of two components. One is a generator to take care of modifications in the real images, and the other is a discriminator that focuses on the wrong assumptions made by the model in the identification process. The basic element needed by any GAN to generate fake images is "noise" to generate different variations in images. However, GANs can accept only random noise or probability distribution. In reality, GAN acts as a mapping function between two different dimensional spaces. Designing a good model requires the dimensionality of sample space to map with latent space. Otherwise, the model may lose all the essential properties in the classification process based on the distance. In general, to convert 1D into 2D, the traditional Vanilla GAN tries to analyze the Gaussian distribution among the random samples, but the model fails after a few training epochs. The proposed research defined the latent space dimension as 100, as the traditional GANs cannot support high dimensional conversion, and also, the dataset consists of images of multiple diseases. Hence, the model needs a GAN that can convert one domain to another, which is the basic phenomenon of the consistent cyclic GAN. CGAN consists of two generators and two discriminators, each consisting of 3 layers of CNN in each component, as shown in Fig. 9.

The proposed research modifies the existing 3-layered architecture of GANs into 4-layered architecture (shown in Fig. 10), with a single generator and discriminator, shown in Fig. 10.

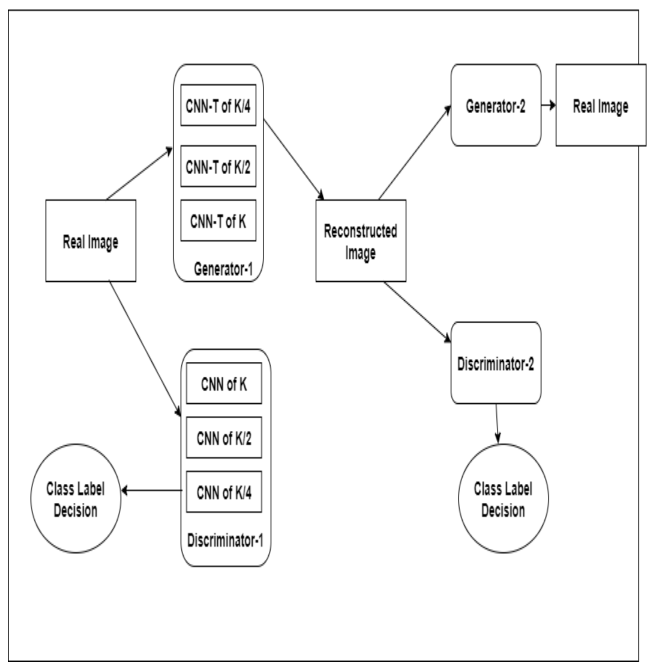

Fig. 9. Existing Architecture of Cycle GAN.

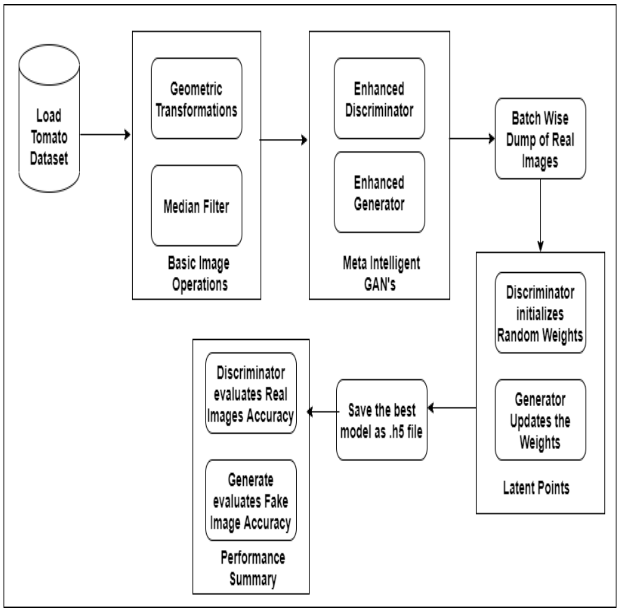

Fig. 10. Complete MI-Leaf GAN Architecture for the Synthetic Data Creation.

Algorithm for MI-Leaf GAN:

Input: T_Data←Load the Pre-processed Images of Tomatoes from the repository
Data: Generates more number of augmented images
Begin:
1. Initialize latent_dspace with 100
2. model_discriminator←call enhanced_discriminator()
3. model_generator←call enhanced_generator(latent_dspace)
4. model_enhanced_CGAN←call enhanced_CGAN(model_generator,model_discriminator)
5. Call train_enhanced_model(model_generator, model_discriminator, model_enhanced_CGAN, T_Data[0],latent_dspace)
6. decision←call generate_latent_points()
7. fake_img←model.predict(decision)
8. Plot the images
End

Pseudocode for Enhanced Discriminator using Sequential Network:
1. Resize the input image to k (where k defines the size as 128)
2. Define four convolution layers of size k/2
3. Define 4 LeakyReLu layers with αn 0.2
4. Add a Flatten layer
5. Add dropout layer to normalize with threshold value as 0.4
6. Add a fully connected layer with sigmoid as an activation function
7. Compile the Model by defining the optimizer, learning rate, loss function, and accuracy

Pseudo code for Training of Enhanced Model
1. Set up number of batches in each epoch i.e., bperepoch as number of images per number of batches
2. Initialize number_of_epochs as "35000"
3. a. for i in number_of_epochs:
b.for j in bperepoch:
c. img_real, label_real←generate_real_images(T_Data,bperepoch/2)
d.img_fake, label_fake←generate_fake_images(model_generator,latent_dspace, bperepoch/2)
e. d_loss←model_discriminator.train_on_batch(img_real, label_real, img_fake, label_fake)
f. img_gan←generate_latent_points(latent_dspace,batch_size)
g. label_gan←np.ones((batch_size,1))
h. g_loss←model_enhanced_CGAN.train_on_batch(img_gan,label_gan)

In binary machine learning algorithms, the class labels will be either 0 or 1, but in GANs, the class labels are assigned as -1, 0, and 1 due to Gaussian distribution. In GANs, the latent space tries to identify the region of interest by performing addition and subtraction operations, i.e., finding the compressed image of the original image by acting as the hidden layer. Each latent space is associated with latent points, which are used to assign random values initially with a discriminator's help. Later, these values are updated through the generator component.

IV. Results and Discussion

The Meta Intelligent Leaf GAN designed using a single discriminator and generator with a constant number of neurons in the hidden layers produces the fake images, as shown in Fig. 11. Since the proposed research uses online data augmentation techniques, these are stored in Google drive for further access.

Fig. 12 shows the accuracy produced by the GAN structure at a different number of epochs. The number of epochs is set to 35000, and the model exhibited an output of 400 epochs as a sample. Similarly, the model can also plot the accuracy rate for the fake images.

To train the GAN structure, the model initially defines the latent space with 100 dimensionalities, and then the pre-processed images are passed as input to the GAN. The proposed model computes the loss functions for both components. Fig. 13 presents a sample output for a single iteration with the corresponding accuracy of real and fake images.

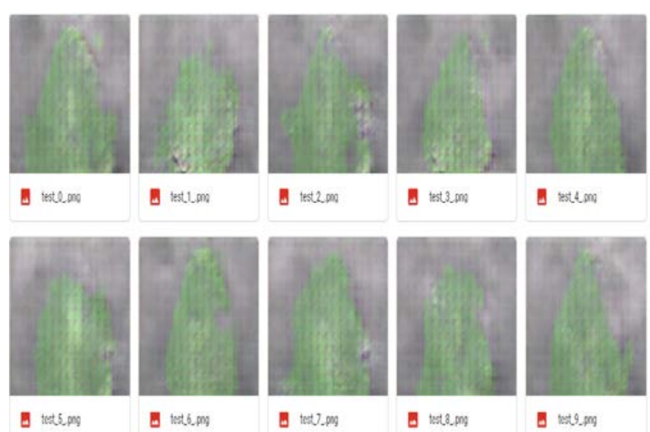

Fig. 11. Synthetic Images Created using MI-GAN.

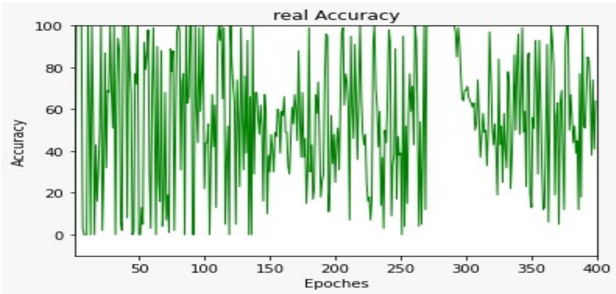

Fig. 12. Accuracy for Generating Real Images at Different Epochs.

```
Accuracy real: 100%, fake: 49%
21, 1/1, d=0.653, g=0.678
22, 1/1, d=0.624, g=0.660
23, 1/1, d=0.593, g=0.635
24, 1/1, d=0.589, g=0.611
25, 1/1, d=0.634, g=0.587
26, 1/1, d=0.714, g=0.593
27, 1/1, d=0.760, g=0.633
28, 1/1, d=0.744, g=0.684
29, 1/1, d=0.708, g=0.744
30, 1/1, d=0.671, g=0.794
```

Fig. 13. Loss Function Computation for GAN.

To date, many researchers worked on different GAN structures. Most of the problems faced are either due to noise or complexity of the design rather than random noise. The first traditional GAN was developed for the MNIST dataset. For 28*28 images, they have used two generators and discriminators with three layers each. Similarly, Table II represents the other mechanisms with the corresponding accuracy and proves that the proposed model has achieved good accuracy and simple architecture.

From Table II, the GAN approach has crossed approximately 90% out of the traditional CNN and GAN. So the proposed system enhanced the GAN technique using meta intelligent concept and reached 99.29%, which is approximately a 2.29% improvement from previous approaches. Accuracy alone cannot be stated as state of the art. So in Table III, the model presents other metrics of learning approaches.

TABLE II. ACCURACY ANALYSIS BETWEEN EXISTING AND PROPOSED GAN

Reference Number	Algorithm	Accuracy
[1]	CNN Integrated GAN	97
[3]	Feature Fusion	78.80
[5]	CNN	97.1
[7]	RPN for Image Segmentation and GAN	83.5
[9]	RAHC-GAN	93.28
Proposed Work	MI-GAN	99.29

TABLE III. EVALUATION METRICS

Algorithm	Accuracy	Precision	Recall	F1-Score
CNN Integrated GAN	97	90	85	87
Feature Fusion	78.8	73	76	74.5
CNN	97.1	90	85	88
RPN for Image Segmentation and GAN	83.5	80	80	80
RAHC-GAN	93.28	85.1	89	87
MI-GAN	99.29	96	98	97

The visualization graph of the Table III analysis is presented in Fig. 14, where the X-axis represents the algorithm's name and the Y-axis represents the measurement scale. The proposed system has performed accurately in all metrics. So, the state-of-the-art for the proposed system is achieved.

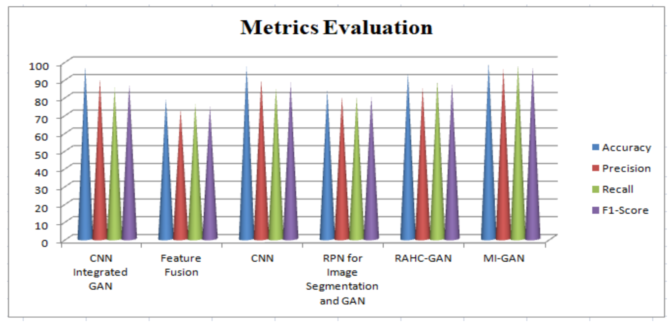

Fig. 14. Metrics Evaluation Analysis.

V. CONCLUSION

It is complicated to label every image from the live capturing. It is difficult to predict the type of disease using unsupervised algorithms like clustering. So, the model should develop a classification system where data labeling is mandatory. In a real-time environment, the conditions are uncontrolled. The method may face problems like labeling disease datasets, addressing the data imbalances, and overfitting because of the size of diagnostic clues, which are smaller than in the real-world scenario. The system also suffers from overfitting because it takes many computations to select and apply possible geometric transformations. Designing a GAN structure for high dimensionality images is challenging because the standard MNIST uses three-layered architecture for simple 28*28 size images. If the size increases, the number of layers get increased, and the model becomes complicated. So, by designing a good policy schema, like Meta Intelligent Leaf GAN, the auto augmentation process can efficiently improve the accuracy where the enhanced GAN model uses an integrated filter. The MI Leaf GAN modules generate more training data because the system addresses the overfitting problem that occurs due to basic image operations. Fig. 13 shows that accuracy is more accurate for the proposed research in generating fake images and successfully fooling the discriminator after performing more epochs. Even though the size of the images is vast, the model considered is more significant than standard GAN; it was designed only in four layers to complete the task.

REFERENCES

[1] Ahmed, S., Hasan, M., Ahmed, T., Sony, R. K., &Kabir, M. (2021). Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification. arXiv preprint arXiv:2109.02394.https://arxiv.org/abs/2109.02394.

[2] Atila, Umit & Ucar, Murat & Akyol, Kemal & Uçar, Emine. (2021). Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics. 61. 101182. 10.1016/j.ecoinf.2020.101182.

[3] Deng, H.; Luo, D.; Chang, Z.; Li, H.; Yang, X. RAHC_GAN: A Data Augmentation Method for Tomato Leaf Disease Recognition. Symmetry 2021, 13, 1597. https://doi.org/10.3390/sym13091597. https://www.mdpi.com/2073-8994/13/9/1597.

[4] Wu, Y.; Xu, L. Image Generation of Tomato Leaf Disease Identification Based on Adversarial-VAE. Agriculture 2021, 11, 981. https://doi.org/10.3390/agriculture11100981. https://www.mdpi.com/2077-0472/11/10/981.

[5] Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 187, 106279. doi:10.1016/j.compag.2021.106279 . https://www.sciencedirect.com/science/article/abs/pii/S0168169921002969.

[6] Karnik, Jashraj and Suthar, Anil, Agricultural Plant Leaf Disease Detection Using Deep Learning Techniques (June 27, 2021). Proceedings of the 3rd International Conference on Communication & Information Processing (ICCIP) 2021, Available at SSRN: https://ssrn.com/abstract=3917556 or http://dx.doi.org/10.2139/ssrn.3917556.

[7] Aaditya prasad, Nikhil Mehta, Matthew Horak, and wan d. bae. A Two-Step Machine Learning Approach For Crop Disease Detection: An Application Of GAN and UAV Technology. (Sep 19, 2021), consideration in the International Journal on Artificial Intelligence Tools. https://arxiv.org/abs/2109.11066.

[8] Bhattacharya S., Mukherjee A., Phadikar S. (2020) A Deep Learning Approach for the Classification of Rice Leaf Diseases. In: Bhattacharyya S., Mitra S., Dutta P. (eds) Intelligence Enabled Research. Advances in Intelligent Systems and Computing, vol 1109. Springer, Singapore. https://doi.org/10.1007/978-981-15-2021-1_8.

[9] Guo, Yan; Zhang, Jin; Yin, Chengxin; Hu, Xiaonan; Zou, Yu; Xue, Zhipeng; Wang, Wei (2020). Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming. Discrete Dynamics in Nature and Society, 2020(), 1–11. doi:10.1155/2020/2479172 .

[10] Faye Mohamed, Chen Bingcai, Kane Amath Sada; Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village, *Journal of Computer and Communications, Vol.8 No.6, 2020; DOI:* 10.4236/jcc.2020.86002 .

[11] Ayaluri MR, K. SR, Konda SR, Chidirala SR. 2021. Efficient steganalysis using convolutional auto encoder network to ensure original image quality. PeerJ Computer Science 7:e356 https://doi.org/10.7717/peerj-cs.356.

[12] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* **6,** 60 (2019). https://doi.org/10.1186/s40537-019-0197-0.

[13] A Mallikarjuna Reddy, Vakulabharanam Venkata Krishna, Lingamgunta Sumalatha and Avuku Obulesh, "Age Classification Using Motif and Statistical Features Derived On Gradient Facial Images", Recent Advances in Computer Science and Communications (2020) 13: 965. https://doi.org/10.2174/2213275912666190417151247.

[14] Maniyar H.M., Budihal S.V. (2020) Plant Disease Detection: An Augmented Approach Using CNN and Generative Adversarial Network (GAN). In: Badica C., Liatsis P., Kharb L., Chahal D. (eds) Information, Communication, and Computing Technology. ICICCT 2020. Communications in Computer and Information Science, vol 1170. Springer, Singapore. https://doi.org/10.1007/978-981-15-9671-1_22.

[15] Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Aboul Ella Hassanien, Hari Mohan Pandey, An optimized dense convolutional neural network model for disease recognition and classification in corn leaf, Computers and Electronics in Agriculture, Volume 175, 2020, 105456, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2020.105456.

[16] Ilaiah Kavati, A. Mallikarjuna Reddy, E. Suresh Babu, K. Sudheer Reddy, Ramalinga Swamy Cheruku,Design of a fingerprint template protection scheme using elliptical structures,ICT Express,Volume 7, Issue 4,2021,Pages 497-500,ISSN 2405-9595,https://doi.org/10.1016/j.icte.2021.04.001.

[17] Xiao, Jia-Rong; Chung, Pei-Che; Wu, Hung-Yi; Phan, Quoc-Hung; Yeh, Jer-Liang A.; Hou, Max T.-K. 2021. "Detection of Strawberry Diseases Using a Convolutional Neural Network" *Plants* 10, no. 1: 31. https://doi.org/10.3390/plants10010031.

[18] C. R. T, G. Sirisha and A. M. Reddy, "Smart Healthcare Analysis and Therapy for Voice Disorder using Cloud and Edge Computing," *2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Mangalore, India, 2018, pp. 103-106, doi: 10.1109/iCATccT44854.2018.9001280.

[19] Rakesh Chandra Joshi, Manoj Kaushik, Malay Kishore Dutta, Ashish Srivastava, Nandlal Choudhary, VirLeafNet: Automatic analysis and viral disease diagnosis using deep-learning in Vigna mungo plant, Ecological Informatics, Volume 61, 2021, 101197, ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2020.101197.

[20] Padmanabhuni, Ms. S. S. (2020). AN EXTENSIVE STUDY ON CLASSIFICATION-BASED PLANT DISEASE DETECTION SYSTEMS. JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES, 15(5). https://doi.org/10.26782/jmcms.2020.05.00002

[21] Wu, Q., Chen, Y., & Meng, J. (2020). DCGAN-Based Data Augmentation for Tomato Leaf Disease Identification. IEEE Access, 8, 98716–98728. doi:10.1109/access.2020.2997001.

[22] A. M. Reddy, K. SubbaReddy and V. V. Krishna, "Classification of child and adulthood using GLCM based on diagonal LBP," 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, 2015, pp. 857-861, doi: 10.1109/ICATCCT.2015.7457003.

[23] Mohana Saranya, S., Rajalaxmi, R. R., Prabavathi, R., Suganya, T., Mohanapriya, S., &Tamilselvi, T. (2021). Deep Learning Techniques in Tomato Plant – A Review. Journal of Physics: Conference Series, 1767(1), 012010. doi:10.1088/1742-6596/1767/1/012010 https://iopscience.iop.org/article/10.1088/1742-6596/1767/1/012010/meta.

[24] Swarajya Lakshmi V Papineni, Snigdha Yarlagadda, Harita Akkineni, A. Mallikarjuna Reddy. Big Data Analytics Applying the Fusion Approach of Multicriteria Decision Making with Deep Learning Algorithms *International Journal of Engineering Trends and Technology*, 69(1), 24-28, doi: 10.14445/22315381/IJETT-V69I1P204.

[25] Abd El-Latif, Yasser. (2021). Early Prediction of Plant Diseases using CNN and GANs. International Journal of Advanced Computer Science and Applications. 12. 514-519. 10.14569/IJACSA.2021.0120563.

[26] PlantVillage Dataset Repository: https://github.com/spMohanty/PlantVillage-Dataset.

[27] A. M. Reddy, V. V. Krishna, L. Sumalatha and S. K. Niranjan, "Facial recognition based on straight angle fuzzy texture unit matrix," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, 2017, pp. 366-372, doi: 10.1109/ICBDACI.2017.8070865.

[28] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 59069-59080, 2019, DOI: 10.1109/ACCESS.2019.2914929.