Generalized Anomaly Recognition and Detection for Enhanced Nurturing (GARDEN)

David Xia

University of Illinois Urbana-Champaign

Department of Electrical and Computer Engineering,

College of Design and Engineering

National University of Singapore

Contents

I. Preface and Acknowledgements

This project explores the ubiquitous task of anomaly detection, which is applicable to a wide spectrum of studies and scientific domains. This project proposes and applies an anomaly detection pipeline on diseased plant leaves. The prevalence of plant leaf data, combined with the pressing issue of agricultural sustainability makes plant leaves the perfect subject to investigate. Leaf image data is sourced from online sources as well as local farms, spanning different diseases and defects across multiple species of plants. To explore anomaly detection of plant leaves, this paper proposes and introduces GARDEN: Generalized Anomaly Recognition and Detection for Enhanced Nurturing, a machine learning pipeline for anomaly detection

This project is done in parallel with four other projects also researching anomaly detection for plant leaves. I would like to acknowledge Gloria Kalnitskaya from Johns Hopkins University, Meiling Mathur from the University of Pennsylvania, Gavin Wang from the University of Wisconsin-Madison, and Andrew Zhang from the University of Wisconsin-Madison for their input and support. I also wish to express my gratitude to Dr. Kelvin Fong Xuanyao from the National University of Singapore who supervised the project within the Department of Electrical and Computer Engineering in the College of Design and Engineering of the National University of Singapore. I would also like to recognize the lab members of the Computational Nanoelectronics and Nanodevices Laboratory who assisted and guided the project. Furthermore, I am also grateful for V-Plus Agritech, a local aquaculture farm that provided image data and highlighted the glaring issues in modern agriculture. The experiences and insights from visits to the farm helped guide this project's direction.

II. Abstract

Anomaly detection is a critical aspect of data analysis and machine learning, focusing on identifying patterns in data that do not conform to expected behavior. These unexpected patterns are referred to as anomalies, outliers, novelties, or exceptions, and they can signify important, actionable insights in various contexts. Anomaly detection is a topic that has proven useful in a wide range of fields of research. One such area is monitoring plant leaf health. Early detection of diseases and pests can be essential for maintaining good plant health. When applied to vegetables and farm plants, this can also mean food security and sustainability for many. While these topics can be generalized to any similar application, this paper will focus on and introduce a pipeline for anomaly detection for plant leaves. The Generalized Anomaly Recognition and Detection for Enhanced Nurturing (GARDEN) pipeline utilizes deep learning techniques and begins with a segmentation model that isolates leaves from their background. This allows the rest of the pipeline to focus solely on areas of interest. The pipeline then wields a generative model based on a conditional Generative Adversarial Network (cGAN) architecture. This generative model learns the patterns and features of these leaf images to generate synthetic images. These synthetic images can prove extremely useful in expanding a dataset and introducing diversity. A classification model trained on these real and synthetic images learns to differentiate between diseased and healthy leaves. When used on natural images of leaves in a farm, this classification model can perform early detection of diseases. Stacking this deep learning classification model with traditional machine learning models, the pipeline utilizes the strengths of different models to mitigate their individual weaknesses.

III. Introduction

The world's population is expected to increase by nearly 2 billion within the next 30 years [1]. Compounded with the up to 828 million people who were starving in 2021 [2] and decreases in traditional agricultural yield due to climate change [3], food security has become a serious issue worldwide. One of the proposed alternatives to traditional agricultural techniques is aquaculture. Aquaculture is the controlled cultivation of aquatic life and vegetation. The marine life, most commonly fish, produce nutrients for the plants through their waste and the plants, symbiotically, purify the water for the fish. Some of the many benefits of aquaculture include climate sustainability, food security in non-arable regions, and increased yield [4]. Aquaculture has shown great promise and could be the answer for many starving populations.

However, aquaculture is not without its own difficulties. One of the greatest challenges that not only plagues traditional agricultural techniques, but also aquaculture, is plant health. Especially as pesticides are increasingly stigmatized, pests and their diseases have become a serious challenge for farmers across the world. Studies show that diseased plants significantly lower crop yields and can negatively impact human health [5]. However, to support the booming population growth, farms need to be large and scalable. Traditional techniques of detecting and treating diseased plants by hand becomes expensive and labour-intensive on larger scales.

One of the ways to increase scalability is utilizing machine learning models. Utilizing computer vision techniques, these models can highlight potentially diseased plants. Compared to other disease detection methods like sensors and manual examination, computer vision can be an extremely affordable alternative. However, these models are also not without their own limitations. Traditional machine learning techniques like decision trees, random forests, and logistic regression all lack the required complexity to truly capture the intricacy of these images. On the other hand, deep learning methods require significant amounts of data to train. This is often challenging to source, especially in niche topics.

In response to this issue, this paper will introduce and propose the Generalized Anomaly Recognition and Detection for Enhanced Nurturing (GARDEN) machine learning pipeline. This pipeline consists of multiple models collaborating to improve the overall identification accuracy of real farm images. The first component of the pipeline is the segmentation model, which segments singular images of plant leaves. Then, the generative model is trained using these segmented image cutouts to generate synthetic data. This generative model is based upon a conditional Generative Adversarial Network (cGAN) architecture. These real and synthetic images are then utilized to train the classification model. Another, separate segmentation model is trained and used to segment natural images of clustered leaves in a real farm. The resulting segmented leaves are then isolated and fed into the classification model for anomaly detection. Furthermore, GARDEN also implements stacking with traditional machine learning models like decision trees, random forests, and logistic regression to balance the strengths and weaknesses of both traditional and deep learning techniques.

IV. Background

A. Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) are a class of machine learning frameworks [6]. GANs consist of two neural networks, namely the generator and the discriminator, which are trained simultaneously through an adversarial process. The generator network aims to produce realistic data samples, while the discriminator network evaluates them against real data samples. The generator strives to create data indistinguishable from real data, while the discriminator attempts to distinguish between real and generated data. This competition drives both networks to improve their performance continuously.

The generator starts with a random noise vector and transforms it into a data sample. Initially, these generated samples are of poor quality, but as training progresses, the generator learns to produce increasingly realistic samples. Concurrently, the discriminator learns to become more adept at identifying real and generated samples. The training process is a competition where the generator seeks to minimize the probability of the discriminator correctly identifying fake samples, while the discriminator aims to maximize this probability. The objective function for GANs can be expressed as:

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

where G represents the generator, D represents the discriminator, x denotes real data samples, and z represents the random noise vector input to the generator.

The concept of GANs has since revolutionized the field of generative modeling, leading to significant advancements in various domains such as image synthesis, text generation, and video creation. Numerous GAN variants have been developed to address specific challenges and improve performance, including conditional GANs (cGANs).

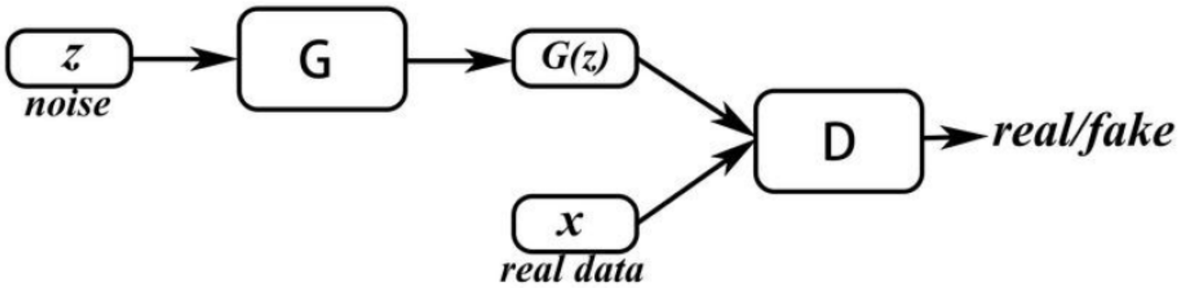


Fig. 1: GAN architecture [7]

B. Conditional GAN (cGAN)

Conditional Generative Adversarial Networks (cGANs) extend the concept of GANs by introducing an additional conditioning information vector, y, into both the generator and discriminator networks [8]. This conditioning can be in the form of class labels, text descriptions, or any other auxiliary information that helps guide the generation process towards desired outputs.

cGANs are particularly useful for tasks where control over the generated output is crucial. The key idea behind cGANs is to condition the generation process on vector y, which enables the generation of specific outputs corresponding to different conditions. For example, in image generation tasks, vector y could represent a class label, allowing the generator to produce images belonging to a specified class.

The objective function of cGANs extends the original GAN objective to include the conditioning vector variable y. The objective function can be formulated as:

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

Here, G represents the generator, D represents the discriminator, x denotes real data samples conditioned on y, z represents the random noise vector input to the generator, and $G(z|y)$ denotes the generated data sample conditioned on vector y.

cGANs have been successfully applied in various domains such as image-to-image translation, image inpainting, and semantic image synthesis. By conditioning the generation process on specific information, cGANs offer greater control and flexibility in generating realistic and contextually relevant outputs. The generative models developed in this project follow the cGAN structure.

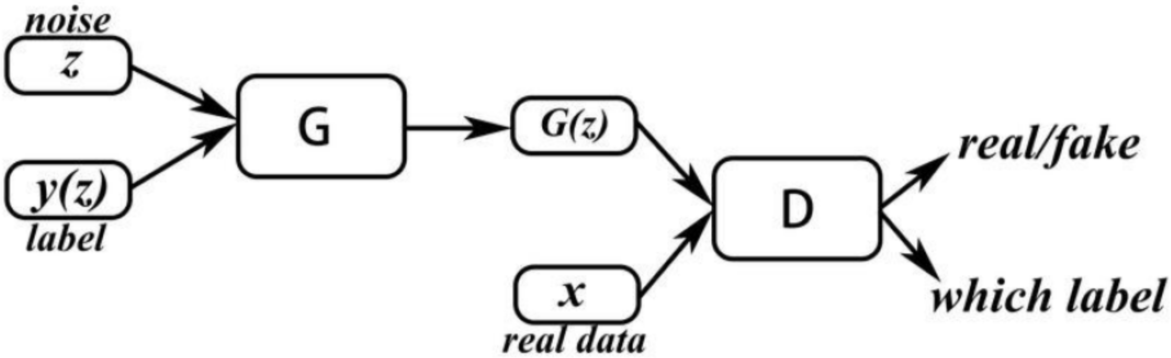


Fig. 2: cGAN architecture [7]

C. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a class of deep learning models particularly well-suited for analyzing visual data [9]. Since its introduction, CNNs have become a cornerstone of modern computer vision

The typical structure of a CNN consists of several layers. These layers all perform different tasks and contribute to the overall feature extraction. The convolutional layers apply convolutional filters to the input image, creating feature maps that highlight various aspects of the image, such as edges, textures, and patterns. The filters slide over the input data, performing element-wise multiplications and summing the results, which helps in detecting spatial hierarchies in the data. Following the convolutional layers, activation functions like ReLU (Rectified Linear Unit) introduce non-linearity into the model, allowing it to learn more complex patterns. Next are the pooling layers, such as max pooling, which reduce the spatial dimensions of the feature maps. This helps control the computational load and mitigate over fitting. Fully connected layers act as a traditional neural network. They take the high-level features extracted by the previous layers and use them to perform a final classification. The output layer generally utilizes a Sigmoid layer for binary classifications or a Softmax layer for multi-class classifications.

CNNs have revolutionized numerous fields through their superior performance in image-related tasks. In this project, CNNs are used for both the segmentation and classification models.

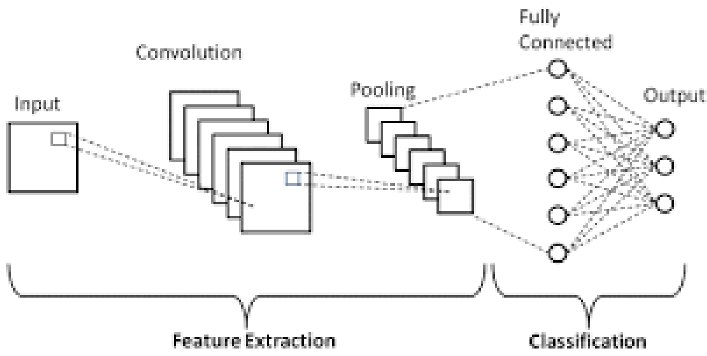


Fig. 3: CNN architecture [10]

V. Methodology

A. Data Collection

1) Singular: Image data is sourced from two different sources in this project. Images of singular plant leaves are sourced from the Kaggle's New Plant Diseases Dataset [11]. This dataset contains a total of around 88,000 singular images of plant leaves. The term 'singular' refers to how the leaf is already separated from the plant in the photo. Below are some examples of how the leaves are represented throughout the entire dataset.

(a) A healthy apple plant leaf

(b) A diseased apple plant leaf

Fig. 4: Examples of leaf images in the dataset

2) Clustered: The other images are sourced from V-Plus Agritech, a local aquaculture farm. Manual images are taken with a smartphone camera. In concept, images would be taken using installed cameras set up on the farm, so the angles of these manual photographs are emulated to be as close as realistic as possible. These images also differ from the previous dataset's images as they are not singular leaves. Early detection of diseases in attached leaves is essential for anomaly control. A total of 200 photograph images are captured. Below are some example images.

Fig. 5: Examples of images from the farm

In this paper, the Kaggle dataset is referred to as the Singular Dataset and the manually photographed dataset is referred to as the Clustered Dataset.

B. Data Processing

1) Singular: The Singular Dataset is already in the desired 256 by 256 .jpg format, so no processing steps are required for this dataset.

2) Clustered: The Clustered Dataset images are originally 3024 by 4032 .heic images. The images are first converted to .jpg format. Then, to reduce the size of the images and shape them to be square, two steps are taken. The first step is creating 100 random 1024 by 1024 patches of each individual image. This not only creates more data for the models to use, but this also focuses each image onto a smaller patch for greater detail. Then, each patch is scaled down from 1024 by 1024 to 256 by 256, and a random rotation is applied. This allows the models to train faster with less memory usage. From the original 200 clustered images, 20,000 patches are created in the desired 256 by 256 .jpg format.

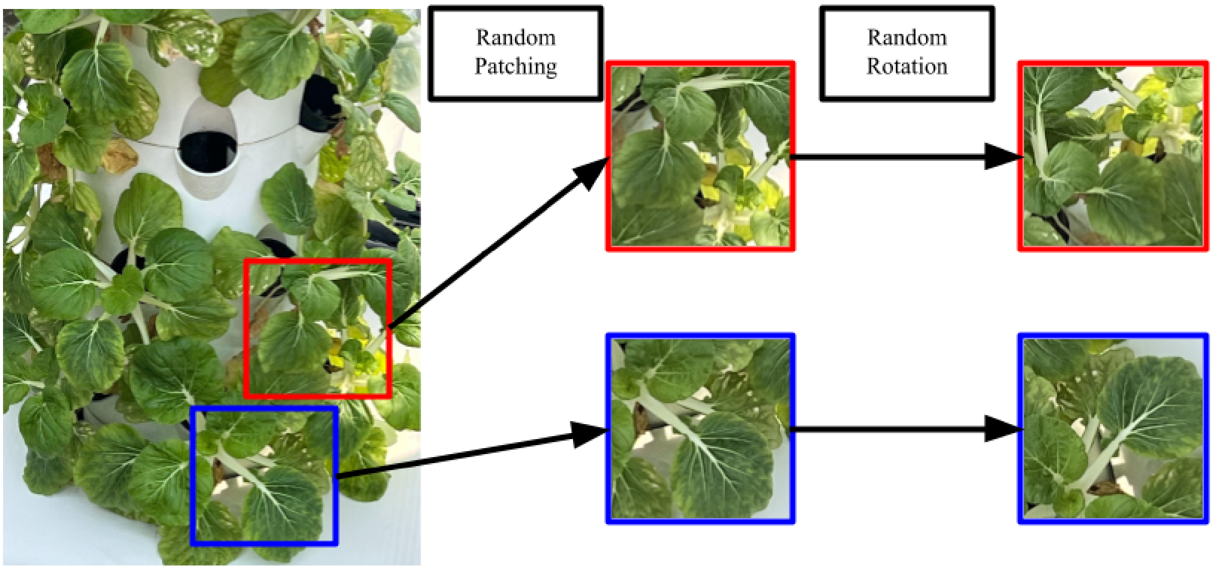


Fig. 6: Processing of clustered images

3) Augmentation: To increase the size of the dataset and introduce more variation into the images, random augmentations are done to create an augmented version of each image. The augmentations are randomly chosen from the Albumentations library and applied to each image [12]. This creates a parallel augmented version of the original dataset. Augmented images are only created for the Singular Dataset. Below are the augmented versions of the examples in Fig. 4.

(a) Augmented version of Fig. 4 (a)

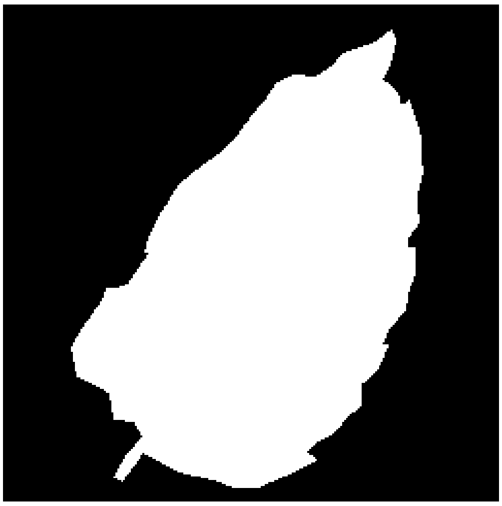
(b) Augmented version Fig. 4 (b)

Fig. 7: Augmented versions of Fig. 4 examples

C. Manual Segmentation

1) Singular: 1000 images (500 unaugmented and 500 augmented) of the Singular Dataset are randomly sampled and segmented. Segmentation is done through the Labelme application. Each image has one polygon drawn outlining the leaf. Each polygon varies between 25 to 75 vertices depending on the complexity and shape of the leaf.

(a) Singular leaf



(b) Manually segmented mask

Fig. 8: Singular image and its mask

2) Clustered: Of the 20,000 patch images in the Clustered Dataset, 600 are randomly sampled to be manually segmented. This is done similar to the manual segmentation of the Singular Dataset. However, because of the overlapping nature of the clustered leaves and the limitations of binary classification, overlapping leaves are segmented as a single leaf. Each manually segmented image consists of 1 - 5 independent leaves.

(a) Clustered leaves

(b) Manually segmented mask

Fig. 9: Clustered image and its mask

D. Automatic Segmentation

1) Singular: The segmentation model is trained using these manual masks. Within the Singular Dataset, models are either trained on solely unaugmented images or both unaugmented and augmented images. Below is a description of the architecture of these models.

The model features a convolutional block that applies two convolutional layers, each followed by batch normalization and ReLU activation. This block learns the more complex features. The encoder block includes a convolutional block followed by a max pooling layer, which halves the spatial dimensions. This block down samples the input while capturing important features. The decoder block performs up sampling using a transposed convolutional layer, which concatenates the resulting feature maps with the corresponding encoder feature maps, and then applies a convolutional block. This helps in recovering the spatial dimensions and refining the feature maps.

The U-Net is constructed by stacking multiple encoder blocks, followed by a bottleneck layer, and then multiple decoder blocks. The input is passed through the encoder blocks, which progressively down sample the feature maps while capturing context. The bottleneck layer connects the encoder and decoder, providing the most compressed representation of the input. The decoder blocks then progressively up sample the feature maps, combining them with the corresponding feature maps from the encoder via skip connections. The final layer applies a convolutional layer with a Sigmoid activation function for binary classification.

2) Clustered: The exact same architecture from the singular segmentation models is used for clustered segmentation models. However, because there may be multiple clusters per mask, each cluster that meets a size threshold of 5000 pixels is then isolated, resized, and pasted onto a black 256 by 256 canvas. This allows for each cluster to be evaluated individually in the classification model.

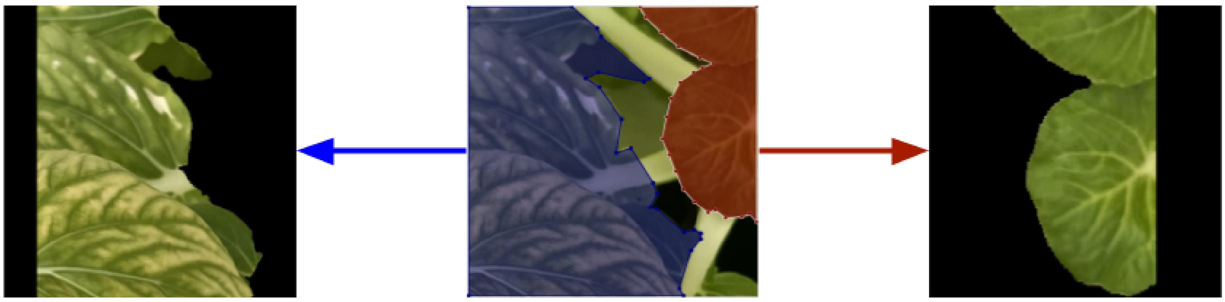


Fig. 10: An example of cluster isolation

E. Generative Model

The generative model is based upon a conditional GAN (cGAN) model. The generator follows a deep convolutional architecture designed to transform random noise vectors into realistic images conditioned on class labels.

The input layer takes a pure noise input. This random noise vector serves as the latent space from which the generator learns to generate images. The class labels are embedded into a dense representation using an embedding layer. This allows the generator to generate synthetic images of specific classes. The generator consists of several convolutional blocks, each designed to progressively up sample the input noise and class information into higher-resolution image features by employing convolutional transpose and batch normalization layers. The activation function is ReLU. The output layer maps the higher-dimensional features into the original image dimensions. Then, the output is passed through a Tanh activation function to ensure pixels are within a valid range.

F. Classification Model

The binary classification model identifies and differentiates healthy and diseased leaves using convolutional neural networks (CNNs). Below is the description of the architecture.

The input layer utilizes a convolutional layer with ReLU activation and also uses max pooling. The convolutional block employs repeated convolutional and max pooling layers. They are each followed by ReLU activation for progressively higher-level feature extraction. The features are flattened into a vector for the dense layer. The dense layer is a fully connected layer with 64 neurons and applies ReLU activation. A dropout rate is also applied to prevent overfitting. The output layer of one neuron uses a Sigmoid activation function to produce a binary classification output.

G. Model Stacking

To combine the strengths of different models, the deep learning classification model is stacked with traditional models. These models include decision trees, random forests, and logistic regression. Different combinations of the models are compared based on their accuracies.

VI. Results

A. Singular Segmentation Results

Nine models with varying hyper parameters are trained to segment the singular image leaves. Below is a table showing the different hyper parameters as well as the model's accuracy and a comparison between the different unaugmented versions' performances.

Version	Learning Rate	Epochs	Batch Size	Accuracy
Augmented 1	$1e^{-3}$	25	32	89.42%
Augmented 2	$1e^{-3}$	50	32	91.93%
Augmented 3	$1e^{-4}$	50	32	91.98%
Unaugmented 1	$1e^{-3}$	25	32	94.06%
Unaugmented 2	$1e^{-3}$	50	32	95.20%
Unaugmented 3	$1e^{-3}$	75	32	99.43%
Unaugmented 4	$1e^{-4}$	75	32	98.43%
Unaugmented 5	$1e^{-4}$	50	32	94.77%
Unaugmented 6	$1e^{-4}$	100	32	98.12%

TABLE I: Singular segmentation model hyper parameters and accuracies

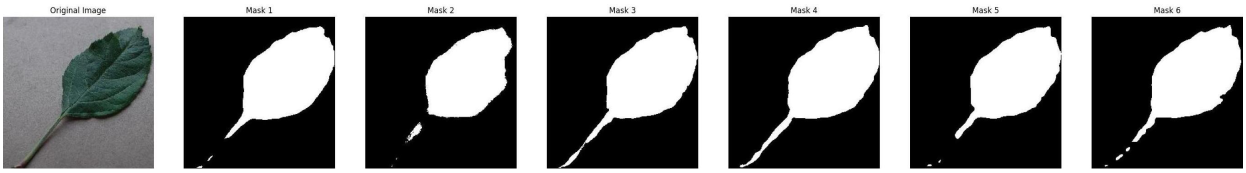
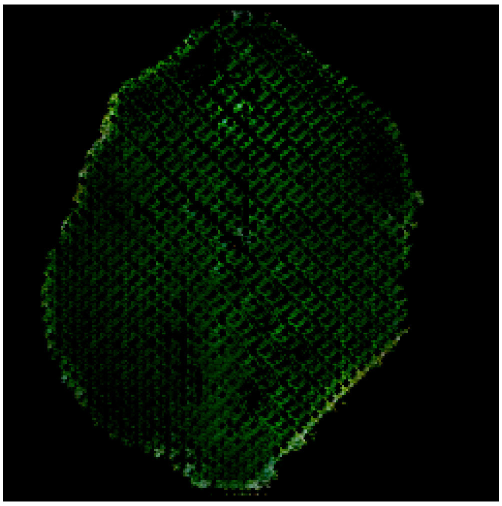


Fig. 11: An example of the different unaugmented versions' performances

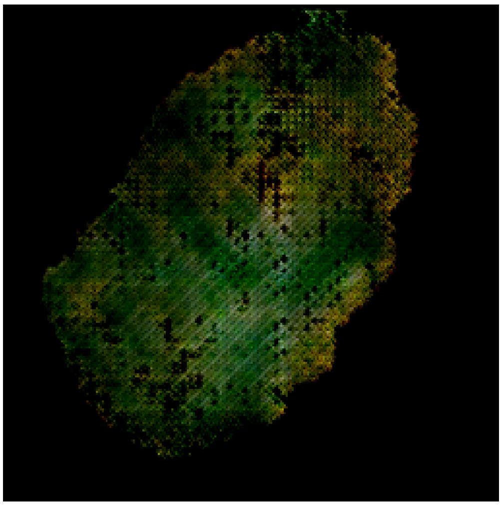
From the table's accuracies and Fig. 11, it's clear to see that the models do remarkably well almost regardless of intense fine tuning. The unaugmented models which are trained exclusively on unaugmented images performed slightly better than the models trained on both augmented and unaugmented images. Version 3 of the unaugmented model does show the best performance and its segmented cutouts are used for the generative and classification models.

B. Image Generation Results

The generative models train using a learning rate of $3e^{-4}$, a low batch size of 8 to prevent memory overload, a latent dimension vector of size 100, and varying epochs. Some results of the models training at 2000 epochs are displayed below.



(a) Synthetic healthy leaf image



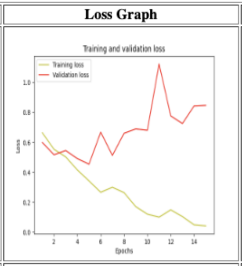
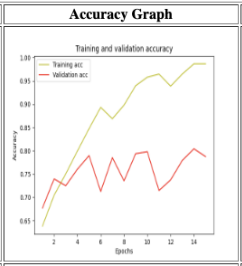
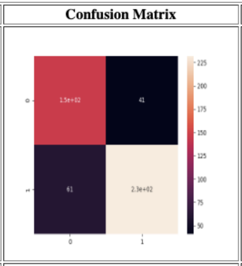
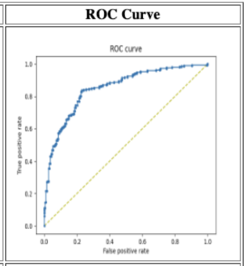
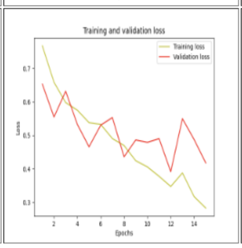
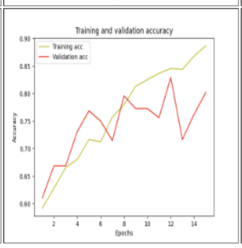
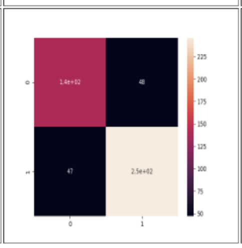
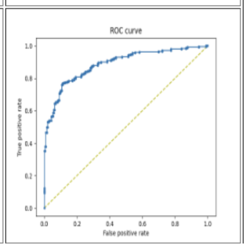
(b) Synthetic diseased leaf image

Fig. 12: Synthetic images

From the synthetic images, it is clear to see they still exhibit too much random noise. This can likely be attributed to the model not fully converging. Allowing the model to continue training will prevent underfitting and generate more convincing results.

C. Classification Results

A total of 128 models are trained to compare accuracies, losses, confusion matrices, and ROC curves. Of these 128, the models training on Version 3 of the unaugmented images perform the best and some of their performances are below.

Ver	Dropout	Optim	BS	Epoch	Loss	Acc	Loss Graph	Accuracy Graph	Confusion Matrix	ROC Curve
v3	0.5	adam	64	15	0.847	0.788				
v3	0.8	rmsprop	64	15	0.417	0.802				

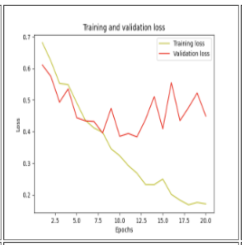
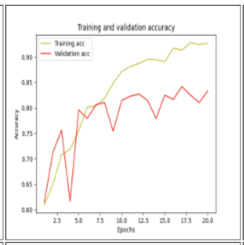
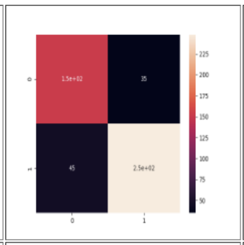
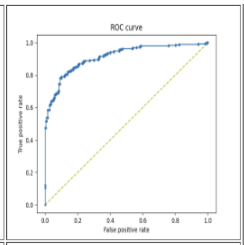
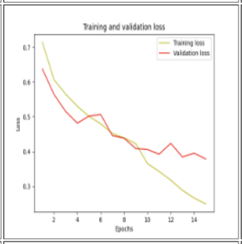
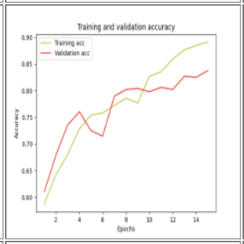
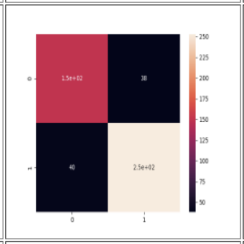
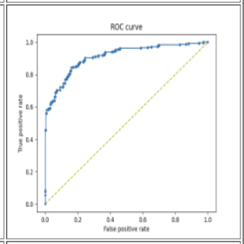
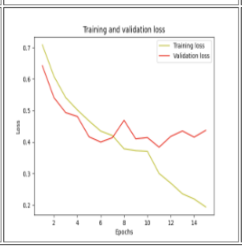
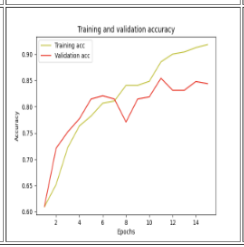
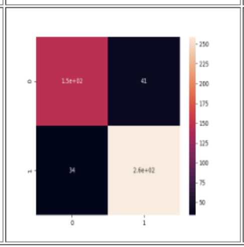
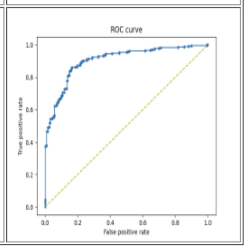
v3	0.8	adam	64	20	0.449	0.833				
v3	0.8	adam	128	15	0.379	0.837				
v3	0.8	adam	64	15	0.437	0.844				

Fig. 13: Unaugmented classification model hyper parameter tuning

The rows are sorted in ascending accuracy with a highest accuracy of 84.4%. Compared to the last model, each of the other four models have an alteration to just one of its hyper parameters. This helps highlight the differences that each hyper parameter contributes.

Multiple dropout rates are tested and a dropout rate of 0.8 is determined to be ideal, which is a rather significant dropout rate. Across all 128 models, the Adam optimizer performed better than the RMSprop optimizer. Furthermore, the loss and accuracy graphs show how training for 20 epochs tends to overfit the model. Different batch sizes are tested, and while there is not any significant difference, a batch size of 64 performed the best. From the confusion matrices, where the false negatives are counted in the top right and the false positives are counted in the bottom left, it is clear to see that the model does well at balancing and does not bias to either side too much.

Below are some selected performances of augmented models.

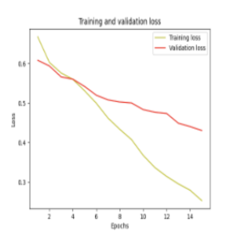
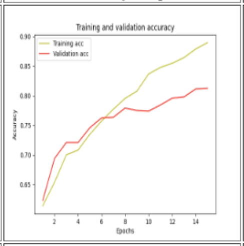
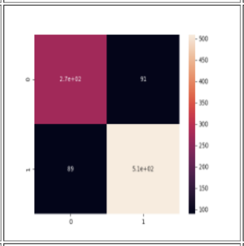
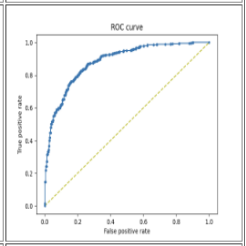
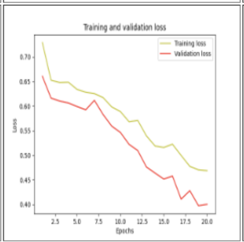
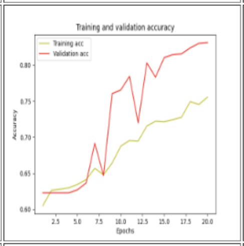
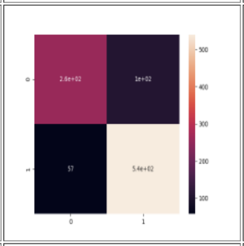
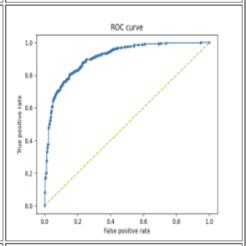
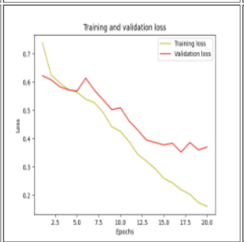
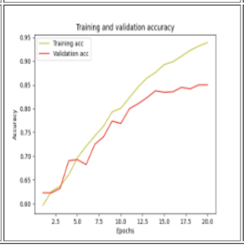
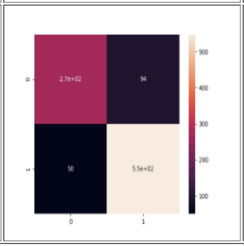
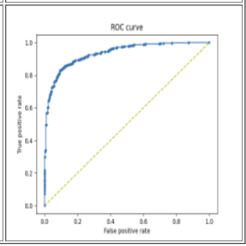
Ver	Dropout	Optim	BS	Epoch	Loss	Acc	Loss Graph	Accuracy Graph	Confusion Matrix	ROC Curve
v3	0.8	adam	128	15	0.431	0.813				
v3	0.95	adam	128	20	0.400	0.831				
v3	0.8	adam	128	20	0.371	0.850				

Fig. 14: Augmented classification model hyper parameter tuning

While the accuracies of the augmented models are comparable to those of the unaugmented models, very high dropout rates are needed to accomplish that feat. Furthermore, from the confusion matrices, the augmented models are a lot less balanced than those of the unaugmented models. Also, to prevent model confusion between augmentations and diseases, only the unaugmented models are chosen for further experimentation and usage.

D. Clustered Segmentation Results

The clustered segmentation model is more complicated than the singular segmentation models. While the singular segmentation models only have to deal with a singular leaf on a blank background, these clustered segmentation models require a deeper understanding to segment multiple, overlapping leaves on a non-trivial background. Below is a table displaying some models and their performances.

Version	Learning Rate	Epochs	Batch Size	Accuracy
Version 1	$3e^{-4}$	15	8	42.88%
Version 2	$3e^{-4}$	50	8	67.41%
Version 3	$3e^{-4}$	100	8	72.32%
Version 4	$3e^{-4}$	200	8	80.90%
Version 5	$3e^{-4}$	75	32	88.85%

TABLE II: Clustered segmentation models hyper parameters and accuracies

From the table, it is clear to see Version 5 is by far the best version and below is an example of an automatically segmented image.

(a) Original image patch (b) Automatically segmented mask (c) Image cutout

Fig. 15: Automatic segmentation of clustered images

The segmentation of these clustered images is quite accurate. However, some of the more complex images face more difficulties in achieving a near perfect segmentation mask. Despite this, a modestly well segmented image is sufficient for the needs of the other models.

E. Model Stacking Results

The deep learning classification model is stacked with multiple traditional machine learning models. Each individual model's prediction is a vote for the final prediction and a meta learner is trained to apply weights to the models' predictions. These traditional models include decision tree, random forest, and logistic regression. Below is a table depicting the performance of different combinations of the models.

Combination	Models				Accuracy
	Decision Tree	Random Forest	Logistic Regression	Deep Learning	
1				●	84.9%
2	●	●	●	●	83.1%
3		●		●	82.9%
4			●	●	81.0%
5	●			●	80.2%
6	●	●		●	78.3%
7		●	●	●	78.3%
8	●		●	●	72.1%
9	●		●		69.0%
10			●		62.6%
11	●	●			62.5%
12	●				62.4%
13	●	●	●		59.9%
14		●	●		58.4%
15		●			58.2%

TABLE III: Combinations of Models and Their Accuracies

The rows are organized in descending accuracy. It is clear to see that the deep learning model has the best accuracy of the models and combinations that include the deep learning model. Moreover, the traditional models do not improve the accuracy of the classification.

VII. Discussion

A. Results Discussion

1) Singular Segmentation: The performance of the Singular segmentation models are remarkably strong. Multiple models boast accuracies ranging from 95% to 99% and the visual examination and comparison of masks reaffirms that accuracy. However, the nature of the Singular Dataset's isolated and clearly photographed leaves makes this feat very achievable.

2) Clustered Segmentation: The accuracy of the Clustered segmentation models is a more interesting statistic. With accuracies as high as 88%, there is certainly a lot of room for improvement. With the greater volume of image data present for the Clustered Dataset, training segmentation models becomes a very time consuming process. However, further fine tuning of hyper parameters could prove extremely useful in increasing the accuracy.

3) Image Generation: The generative model also could see a lot of improvement. The models have not quite reached full convergence and are still underfitting. However, even the synthetic images generated by underfitted models are very valuable and prove the validity of the method. Something else that the generative model suffers from is mode collapse. The synthetic images lack enough diversity to prove effective in bolstering the data volume.

4) Singular Classification: The Singular Dataset classification models boast accuracies of 80% to 85%. Regardless of whether the models are trained using augmented or only unaugmented images, they all seem to begin overfitting very rapidly. Some models begin overfitting after just a few epochs. This could be due to the lack of diversity shown in the original Singular Dataset. Another possible root of the issue is the relative deepness of the model. With such a high model capacity, it is very possible that the model is learning too much random noise and fluctuation.

5) Clustered Classification: The classification model has yet to be deployed for the Clustered Dataset. While the Clustered Dataset is expansive, it remains unlabeled. Each raw image may have dozens of leaves, and when split into random patches, these patches may still have many leaves. Each leaf's health status is independent of its neighboring leaves. While the automatic segmentation of the Clustered Dataset has isolated individual clusters, there is no automatic way to assign labels to these clusters. To use the classification model trained on the Singular Dataset, the accuracy on the Clustered Dataset must be verified manually through a large sample. To train another classification model specifically designed for these clusters, manual labelling also needs to be done for many clusters.

6) Model Stacking: The stacking of the classification model with traditional machine learning models proved to not be as effective. The classification model already boasted higher accuracies than the traditional models. The stacking of different combinations of the models did not improve the overall accuracy.

B. Future Considerations

1) Augmentation: In this project, augmented versions of images did not reach their full potential. More variations of augmentations could prove extremely useful for diversifying the dataset. Furthermore, utilization of less extreme augmentations could improve accuracy. One of the reasons augmented images are less experimented and utilized in this study is because the more extreme augmentations often resembled real diseases, which could confuse models further down the pipeline.

2) Clustered Segmentation: There is a lot of room for future improvement for the clustered segmentation models. With more time and resources, it is definitely possible to optimize the hyper parameters and raise accuracy well above 95%. Furthermore, considering augmented versions of the Clustered Dataset could prove instrumental to boosting the accuracy. Augmented versions of the images could introduce the diversity needed to allow the models to truly capture the underlying patterns.

3) Image Generation: The generative models have a lot of potential in the future. Allowing the models to train for a significant amount of time could show the eventual convergence that is never realized by these early models. The synthetic images generated are also very repetitive and suffer from mode collapse. The addition of augmented images could similarly introduce the diversity the models lack currently. Furthermore, mode collapse is a very understood and widespread issue relating to GANs. Once the generator forges an image that can fool the discriminator, it has very minimal incentive to continue improvising. A very interesting approach would be to try alternative image generation models like diffusion models and variational auto encoders (VAEs), and then compare the results to determine if the underlying issue is the model itself.

4) Singular Classification: The classification models already have a decent accuracy rate and can serve as a good starting base for future growth. Its tendency to overfit could be derived from its architecture, so simplifying the model can be very beneficial. Furthermore, the model could be suffering from a lack of data, which can be resolved through incorporating synthetic images. With improved generative techniques, new synthetic images can bolster the data volume.

5) Clustered Classification: To truly test and validate the strength of the pipeline, manual labelling must be done for the isolated segmentations of the Clustered Dataset. Whether a new model is trained specifically for the Clustered Dataset or the already existing classification model trained using the Singular Dataset is used, manual labelling of thousands of these cluster images needs to be completed. From there, a baseline accuracy can be determined and improved upon.

6) Model Stacking: To improve upon the disappointing results of traditional model stacking, further hyper parameter tuning can be done to these models. These traditional models are only experimented with in a very limited setting, and additional models could also be added to the stack. Training a better meta learner could also benefit the accuracy.

7) Multi-Class Classification: Another improvement that can be made in the future is changing the model from a binary classification to a multi-class classification. From experiences with local farmers, sometimes understanding the root cause of the disease is equally, if not more, important than simply detecting the presence of a disease. Understanding the plant's health holistically and recognizing certain nutrient deficiencies can be vital for plant nurturing.

VIII. Conclusion

The research presented in this paper explored the applications and limitations of segmentation, generative, and classification machine learning models. The segmentation models need some more fine tuning and expansion to include augmentations. The generative models require more research into the architecture and other model types. The classification model, the overall optimization variable in this project, will reap the benefits of the improvements made to the other models. These findings are a strong base to expand and build upon. Overall, the pipeline has displayed the concepts can be incorporated well, but the execution of those concepts needs to be fine tuned and improved. Moving forward, there is a lot of possibility and potential in the future of this project. There are plenty of improvements, optimizations, and ideas for future development. Whether it is experimenting with improved augmentations, changing the models for multi-class classification, or researching different generative techniques, the future direction of this project is exciting and bound to lead to new ideas and innovation.

On a larger scale, this project focuses on the sustainability of the planet's food sources. Food security is a growing and life threatening problem for many people, and researching this topic is of vital importance. Alternatively, these concepts and techniques can also be expanded to a multitude of fields. Through this project, I hope to contribute to a larger understanding in the field of machine learning and anomaly detection.

IX. References

[1] United Nations. *Global Issues: Population*. United Nations, 2023. URL: https://www.un.org/en/global-issues/population (visited on 06/18/2024).

[2] World Health Organization. *UN Report: Global Hunger Numbers Rose to as Many as 828 Million in 2021*. World Health Organization, July 2022. URL: https://www.who.int/news/item/06-07-2022-un-report--global-hunger-numbers-rose-to-as-many-as-828-million-in-2021#:~:text=As%20many%20as%20828%20million%20people%20were%20affected%20by%20hunger (visited on 06/18/2024).

[3] Asim Anand. *How Agriculture Is Exacerbating Global Warming*. Spglobal.com, July 2023. URL: https://www.spglobal.com/commodityinsights/en/market-insights/blogs/agriculture/071323-how-agriculture-is-exacerbating-climate-conditions#:~:text=The%20Asian%20Development%20Bank%20reported (visited on 06/18/2024).

[4] *1. What Are the Main Benefits of aquaculture? — EU Aquaculture Assistance Mechanism*. aquaculture.ec.europa.eu. URL: https://aquaculture.ec.europa.eu/faq/1-what-are-main-benefits-aquaculture (visited on 06/18/2024).

[5] Jean B. Ristaino et al. "The Persistent Threat of Emerging Plant Disease Pandemics to Global Food Security". In: *Proceedings of the National Academy of Sciences* 118 (May 2021), e2022239118. DOI: 10.1073/pnas.2022239118. URL: https://www.pnas.org/doi/10.1073/pnas.2022239118 (visited on 06/18/2024).

[6] Ian Goodfellow et al. *Generative Adversarial Nets*. 2014. URL: https://arxiv.org/pdf/1406.2661 (visited on 06/18/2024).

[7] MK Gurucharan. *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*. UpGrad Blog, Dec. 2020. URL: https://www.upgrad.com/blog/basic-cnn-architecture/ (visited on 06/18/2024).

[8] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. URL: https://arxiv.org/pdf/1411.1784 (visited on 06/18/2024).

[9] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[10] Feilong Han et al. "A Modified Generative Adversarial Nets Integrated With Stochastic Approach for Realizing Super-Resolution Reservoir Simulation". In: *IEEE Transactions on*

Geoscience and Remote Sensing PP (Oct. 2019), pp. 1–12. DOI: 10.1109/TGRS.2019.2945946.

[11] Samir Bhattarai. *New Plant Diseases Dataset*. www.kaggle.com, 2018. URL: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset (visited on 06/18/2024).

[12] A. Buslaev et al. “Albumentations: fast and flexible image augmentations”. In: *ArXiv e-prints* (2018). eprint: 1809.06839.

X. Appendix

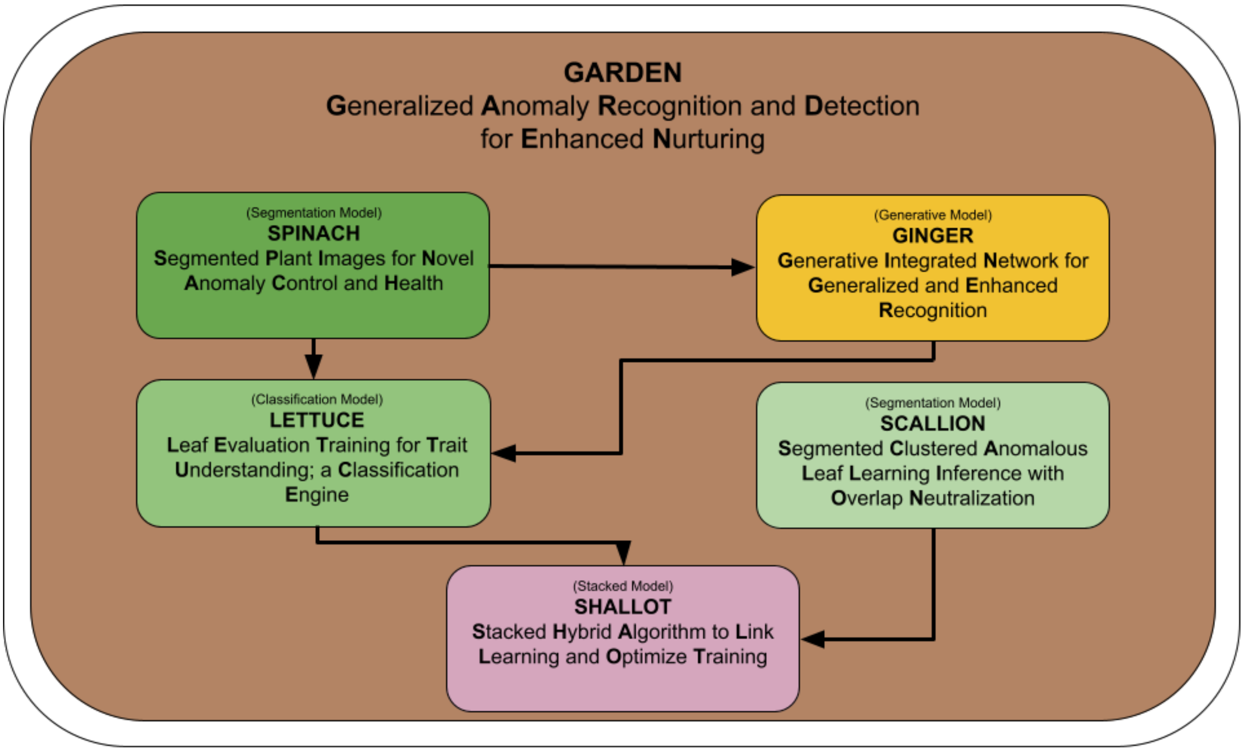


Fig. 16: GARDEN