Plant Disease Detection & Classification Project Report

Machine Learning

UFCFAS-15-2

Group-N-Members   
Ahmed Elsaman - 21072727

Hatim Shaherawala - 21054059

Tommy Diclaudio - 21035734

# Contents

[1. Contents 2](#_Toc134191446)

[2. Abstract 2](#_Toc134191447)

[3. Introduction: 2](#_Toc134191448)

[4. Related work: 3](#_Toc134191449)

[5. Data: 3](#_Toc134191450)

[6. Methodology: 4](#_Toc134191451)

[Planning 4](#_Toc134191452)

[Approach 4](#_Toc134191453)

[Other Approaches 5](#_Toc134191454)

[7. Experiments: 5](#_Toc134191455)

[Basis 5](#_Toc134191456)

[First Model CNN only 6](#_Toc134191457)

[Second model SVM 6](#_Toc134191458)

[Third Model 6](#_Toc134191459)

[Challenges / Failures faced 7](#_Toc134191460)

[8. Reflection 7](#_Toc134191461)

[9. Conclusion 8](#_Toc134191462)

[10. Appendix 8](#_Toc134191463)

[11. References: 9](#_Toc134191464)

# 

# Abstract

*Plant classification and disease detection can be a challenging situation to tackle and requires the system to use techniques such as deep learning to train on the dataset provided. In this project, the problem we are approaching describes infectious diseases affecting crop yields in the agricultural industry where the yields are drastically reduced due to this invasive spread of disease. A Convolutional Neural Network is initiated, which is then developed, tuned, and optimized in extensive experimentation to outperform its initial performance. The PlantVillage dataset is used here, providing 160,000 images with roughly 53,000 in a 3-way split in color, grayscale, and segmented formats. Supporting this, CNN is developed through using VGG16 architecture with transfer learning in its final approach towards optimal performance. Our results demonstrate that the system can predict, in confidence, the classification of the crop as well as if it has been infected and by what disease or if it is in a healthy state.*

# Introduction:

In the field of botany, plant classification is a crucial task that involves categorizing plants based on their characteristics. The conventional method of plant classification relies on manual observation and analysis by experts, which can be time-consuming and subjective. In recent years, machine learning has emerged as a reliable tool to identify plants, as it can provide better, and more accurate results compared to the manual way of classifying plants.

This project aims to develop a plant classification program using machine learning algorithms in a more directed approach. The problem we are approaching is the crop yield problem where a vast amount of crop yields is affected yearly by the crops becoming infected, and then spreading its infection to other crops, which cannot be used. To tackle this, we will develop an algorithm that is not only able to classify these plants, but can compare the images against healthy versions, detecting if the plant has been infected, and identifying what disease infected it. This will in turn help our chosen problem as the system can be used in agriculture where infected crops are identified early, where infection to the rest of the yield can be reduced. Measures can further be taken to mitigate the threat by understanding what type of disease is present (suggested by the system) and putting methods in place to prevent it coming back. From this, the aim of the system is to support farmers and increase the threatening crop yield deduction from occurring, by avoiding the infectious spread.

The success of this project relies on the ability for our system to be able to identify what plant confidently and successfully is what, and whether it is healthy or what disease Is infecting it. This success was found through our results in extensive experimentation, where the program achieves a high accuracy and confidence rate.

# Related work:

Initially, we reviewed lecture materials to understand the concept of convolution neural networks (CNNs). Our research led us to the Plant Village Disease Classification | Acc: 99.6% (Kaggle,2023), where we learned about several concepts such as Discrete Wavelet Transform (DWT), Gray-Level Co-occurrence Matrix (GLCM), and Principal Component Analysis (PCA). Although these concepts were intriguing, we found it challenging to comprehend them fully. Therefore, we decided to conduct further research to gain a better understanding of these concepts.

Next, we read a paper where the researchers conducted a study where they developed a Convolutional Neural Network (CNN) to automatically identify plant species using mobile imagery from plants native to Ireland. The dataset was pre-processed through background removal and data augmentation, and several CNN models were evaluated to determine the most effective approach for plant identification. The best-performing model was integrated into a web application, allowing users to classify new plant images. This study not only addressed the challenges of plant species identification but also showcased the potential of combining botany and computer science to advance biodiversity conservation efforts. (Mangina *et al.*, 2022).  
  
Alok Kumar and Ankit Kumar,2023 in their article describe the use of the VGG 16 architecture as a classifier and feature extractor for a plant disease detection system. The experiments were performed on two plant species, potatoes, and tomatoes. The input to the network is a 64x64 RGB image, and the output is a vector of size 4, representing the four classes in the dataset. The article mentions that the dataset has multiple classes, but it does not provide details on the number or type of classes. The article suggests that the deep learning-based architecture achieved significant results, which could be improved by adding more data and experimenting with different optimizers. (Kumar, A., Kumar, A., 2023).

# Data:

The **plantVillage** **dataset**(kaggle,2019) we are using contains 160,000 images of 14 different plant species, divided into three classes: color, grayscale, and segmented. Each folder for a species contains images of healthy plants and those with specific diseases. This dataset helps train the machine learning model to recognize healthy and infected plants' characteristics. Our project aims to use this dataset, obtained from sources such as Kaggle, to develop a machine learning solution that identifies infected plants to treat them and prevent disease spread in crop yields.

Due to the massive size of the plantVillage dataset with 160,000 images, our project faced challenges in handling and processing it. As a result, we had to make the decision to use only the colored version of the data, which included 53,000 images, one-third of the entire dataset.

To use the complete dataset, we experimented with different solutions, including using multiple PCs or leveraging online resources such as Google Colab. However, we encountered several issues that prevented us from utilizing the full dataset. As a result, we had to work with a smaller portion of the dataset to train our machine learning model. Despite this limitation, we were able to achieve satisfactory results and develop a solution to identify infected plants in crop yields.

For our project the first thing we did was to prepare our dataset using the **ImageDataGenerator** tool. We used this tool to generate new images by applying different transformations to them, such as rotating, zooming, and flipping. By doing this, we have created a more diverse data set, which will help improve our model's accuracy.

Next, we split our dataset into three different sets: a training set, a validation set, and a test set. The training set will be used to train our model, the validation set will be used to tune our model's hyperparameters and prevent overfitting, and the test set will be used to evaluate our model's performance on unseen data.

# Methodology:

## Planning

During the initial planning phase of the project, we considered using established methods such as DWT, GLCM, and PCA. However, upon further examination, we realized that our understanding of these methods was incomplete and implementing them would be challenging. As a result, we decided to conduct further research on the plant village dataset problem to determine the best approach to solving it.

## Approach

After a series of trials and errors, we opted for a pre-trained machine learning model known as **VGG16** for our image classification task. This model was already trained on a vast image dataset, which equipped it with the ability to recognize various patterns and features typically found in images.

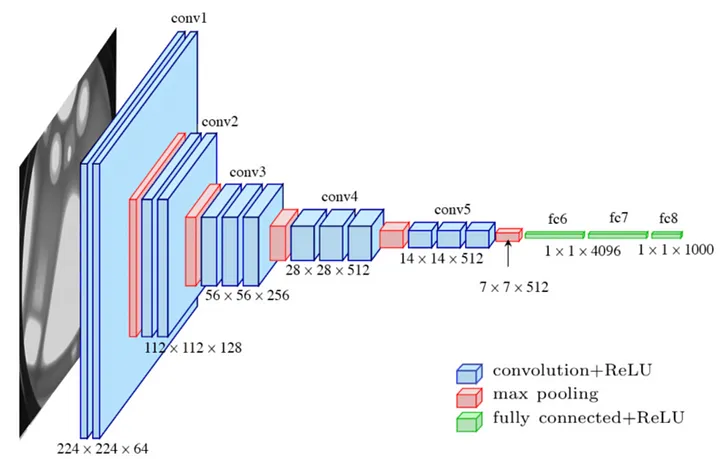


Figure 1: The architecture of VGG16 (Le.K, 2021).

The architecture of VGG16 consists of a series of convolutional layers followed by max-pooling layers, with fully connected layers at the end. The convolutional layers use small 3x3 filters, which helps to capture fine-grained details in the images. The max-pooling layers down sample the feature maps, reducing the spatial dimensions and helping to make the model more efficient.

VGG16 is often used as a base model for transfer learning in image classification tasks, as it has learned a rich set of features that can be used as a starting point for training on a new dataset. The pre-trained weights of the model can be loaded, and then the fully connected layers can be replaced with new layers that are tailored to the specific task at hand. By doing this, the model can be fine-tuned to recognize new classes of objects in images.

We added new layers to the VGG16 model using the K**eras** Functional API. These layers include a Flatten layer, a fully connected Dense layer with 256 units and **ReLU** activation function, and a final Dense layer with 38 units and **softmax** activation function to predict the probabilities for each of the 38 classes.

To improve our model's accuracy, we performed fine-tuning on the last 10 layers of the pre-trained VGG16 model. This allowed us to update the weights of these layers during training and optimize our model's performance for our specific dataset.

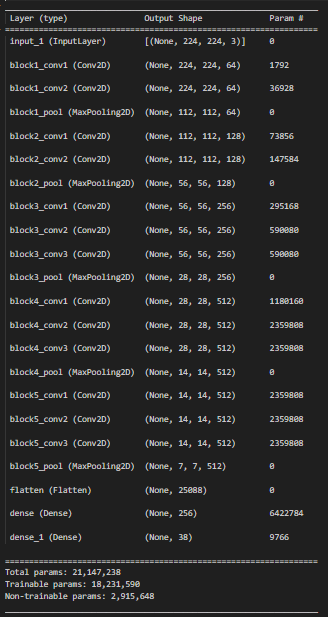


Figure2: Our Model

We then compiled and trained our model using the training and validation generators and evaluated its performance on the test set. We're hoping that our model will be able to accurately predict the different classes of plant diseases, which could be valuable for identifying and treating these diseases in the future.

## Other Approaches

After conducting research and attending lectures, we had first decided to adopt a **sequential CNN** model for our image classification task. This was based on the model's ability to automatically extract useful features from raw image data, eliminating the need for manual feature extraction. The pooling layers in the model aid in reducing the output's spatial dimensions and parameters, thereby enhancing efficiency, and minimizing overfitting. However, despite these advantages, we found that this approach did not perform well in our specific task, as it resulted in **lower confidence rates**.

In addition to our previous experimentation with the model, we attempted to incorporate an **ensemble of SVM and CNN**, with the SVM learning from CNN. This approach was based on our learnings from individual projects.

Like our previous experiment, we used the same properties in the CNN for the SVM model and employed grid search to tune the SVM's hyperparameters to improve its accuracy. However, during our experimentation, we found that **the accuracy** of the SVM model was **not as good** as the normal model we had been using. As a result, we decided to abandon this approach and explore alternative methods for improving the accuracy and confidence of our model.

# Experiments:

## Basis

To address the issue of identifying diseases in thousands of plant images, we conducted numerous experiments with various model architectures, parameters, and tuning methods. Our task is quite specific, involving the classification of healthy and diseased plants. Through **deep learning**, the model is trained to recognize what a healthy plant looks like, enabling it to correctly identify and classify healthy plants. Similarly, the model learns to identify diseased plants by comparing them to their healthy counterparts, and it can differentiate between various diseases based on their distinct visual characteristics.

## First Model CNN only

As a team working with deep learning and image classification, we initially implemented a CNN model due to its effectiveness in automatically learning important features from raw image data. The model was designed to identify patterns and features unique to each image, which would then be used to classify it into one of 38 classes. This was achieved by passing the images through a series of layers, each of which performed operations on the output of the previous layer. We incorporated ReLU activation to improve efficiency and reduce training time. However, we noticed that while the accuracy of the model was decent (Figure 3), the confidence levels were low, indicating a lack of robustness (Figure 4). To address this issue, we explored other techniques such as data augmentation and dropout regularization to improve the generalization performance of the model and prevent overfitting. Our goal was to improve the confidence levels and make the model more robust to variations in the input data.

## Second model SVM

We decided to experiment with the SVM model by combining it with CNN’s learning capabilities as it was a known model to the group that we understood. We utilized the same layer structure in CNN and used the SVM model for predictions. In addition, we employed a grid search to tune the SVM's hyperparameters and improve its accuracy (Figure 5) However, despite extensive tuning efforts, we found that the accuracy of the SVM model was significantly lower than that of the standard CNN model, at around 10% (Figure 6). This clearly shows that the model does not perform well and cannot be trusted. As a result, we decided to abandon this approach and explore other alternatives to improve the confidence and accuracy of the model.

## Third Model

As discussed before, our research into deep learning architectures suitable for our dataset size and problem led us to consider the VGG16 architecture as a potential solution. This architecture has been successful in achieving high accuracy and confidence in other projects. We experimented with **transfer learning**, using **imagenet** as a pre-trained model to improve our model's accuracy. We transferred our classification layer to the last layer of the **imagenet** model, freezing its training so that our model could utilize it. On top of this, hyperparameter tuning and other techniques were implemented to support the model’s accuracy. The tuning included **SGD, learning rate and momentum**. In this part, a neural network is being made for training where the loss function optimizer is being set to be used in the training process. **SGD** trains this network and optimizes by adjusting the weights and minimizing the loss function. The **learning rate** and **momentum** are then passed to this where learning rate defines how much the weights are adjusted, with momentum supporting the model to overcome local minima and instead reach global minimum of the loss function. The **loss function** here was defined **categorical cross-entropy,** which is used to predict categorical outcomes, suitable for our model as the image needs to be picked out of the **38 classes**, belonging to one. This tuning was completed through severe trial and error and experimentation, where ranges of values were compared and the best one was narrowed down to. The values with the best results during our experimentation were found to be a learning rate at 0.001, momentum at 0.9 and the optimizer as SGD. These were used in the final model (Figure 7).

After implementing the architecture and tuning, and a long-running time, our **accuracy improved** (Figure 8) which was a bonus. We then turned our attention to the confidence of the model's predictions as our focus in this improvement. Here we found that when given an image to classify, the model outputs a confidence level of **90-100%** on correctly identified images (Figure 9). This demonstrates our model’s ability in being able to predict the correct choices on whether a plant/crop has been infected and can determine what type of disease this is. Therefore, it can be said that our model approach solves the crop yield problem in being able to identify the infected plants early, to prevent spread in the yield and increase the yield.

Overall, the implementation of the VGG16 architecture with transfer learning and the extensive hyperparameter tuning in our model **proved to be a successful approach** in achieving higher accuracy and confidence levels for our image classification task.

Thus, our final model shows that our approach solves our problem as its accuracy and confidence is at a very stable and high level, demonstrating that the model can predict the correct choices confidently, and can therefore be used in identifying the infected plants in the crop yield problem.

## Challenges / Failures faced.

The size of the dataset can pose a significant challenge when it comes to training deep neural networks on a **normal PC**. With a dataset of **150,000 images**, training such a model can take an incredibly long time, even on a high-end PC. The reason for this is that the training process involves many matrix computations, which require significant computational power. A normal PC might not be equipped to handle such computations and may take a very long time to complete the training process.

Moreover, the choice of deep neural network models also plays a crucial role in the success of the project. We spent a considerable amount of time exploring different models and finally decided to use the VGG16 architecture. However, we soon realized that we did **not have enough computational power to train such a very deep neural network**. Very deep neural networks have multiple layers, which significantly increases the number of parameters to be learned, leading to a higher computational requirement.

To address these issues, we tried various options such as **Google Colab** and running models on **multiple PCs**. However, even with these measures, the training time remained significantly longer, and we ended up using only a subset of the dataset. The size of the dataset was a significant issue, and it required us to make strategic decisions regarding the models and the hardware resources used for training.

# Reflection

Based on the knowledge and experience gained during the project, we are now aware of the limitations we faced and how we can improve our approach in future projects. For instance, we learned that using a normal PC for training a very deep neural network with a large dataset is not feasible and can lead to long processing times, resulting in restricted experimentation and the need to use only a subset of the data. In future projects, we would aim to use powerful hardware or cloud-based services like Google Colab to train our models efficiently.

We also learned the importance of choosing the right model architecture for the problem at hand. In this project, we experimented with different models and found that the VGG16 architecture with transfer learning from the ImageNet dataset gave the best results. However, in future projects, we would explore other architectures that could potentially perform better.

Moreover, we now know that increasing the number of epochs can significantly improve the accuracy of the model. In this project, we only defined one epoch due to hardware constraints, but in future projects, we would try to increase the number of epochs to at least 10.

Overall, based on our learnings from this project, we can significantly improve the performance of our models in future projects by using powerful hardware, choosing appropriate model architectures, and increasing the number of epochs.

# Conclusion

In conclusion, this project successfully demonstrated the application of machine learning, specifically Convolutional Neural Network (CNNs), in the field of plant classification. Through the development of a plant classification program using the VGG16 architecture and transfer learning, the project achieved high accuracy rates. Despite challenges such as the size of the dataset, we managed to achieve satisfactory results by working with a smaller portion of the data. We learned the importance of selecting the appropriate model architecture and making strategic decisions regarding hardware resources.

Also, several possibilities exist for extending this project or developing new applications. One potential extension is to incorporate additional plant species and diseases to create a more comprehensive classification system. This would require using the entire data set and powerful hardware to train the model efficiently. Additionally, the classification system could be integrated into mobile applications to make it a more efficient plant identification system, ultimately benefiting biodiversity conversation efforts and improving crop yields.

# Appendix

Figure 3

Shows first CNN Accuracy

A screenshot of a computer

Description automatically generated with medium confidence

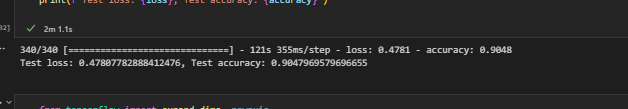


Figure 4

Shows first CNN confidence

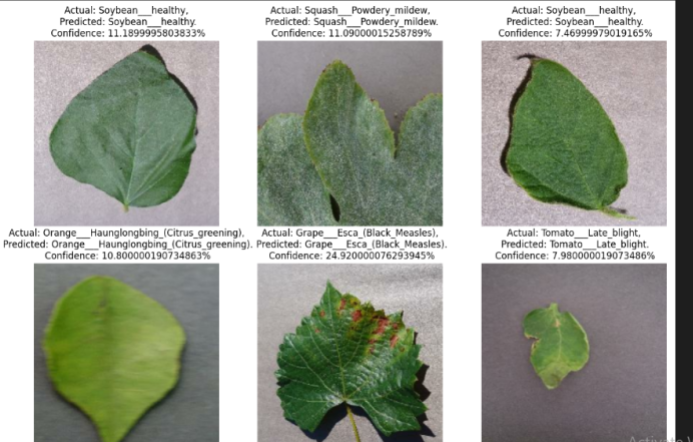


Figure 5

Shows SVM hyper tuning.

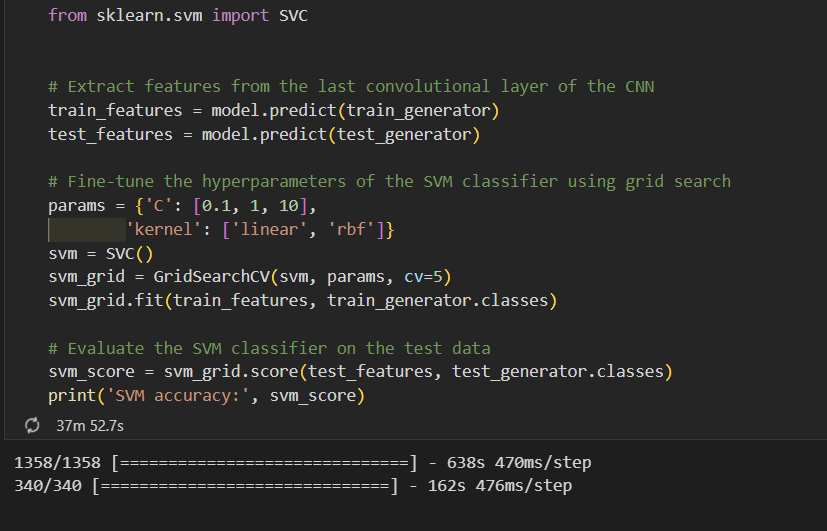


Figure 6

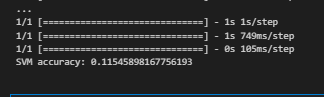
Shows SVM model Accuracy  


Figure 7

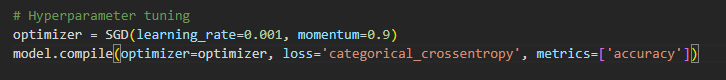


Figure 8

Shows final model accuracy.

A screenshot of a computer

Description automatically generated with medium confidence

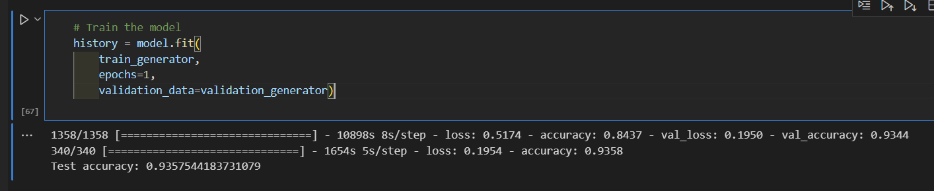
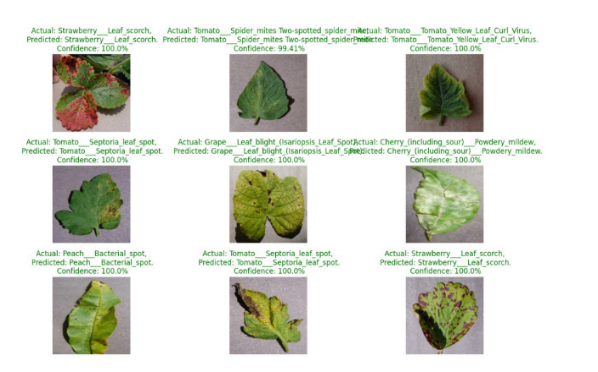


Figure 9

Shows final model confidence.



# 11. References:

1. Kumar, A., Kumar, A. and, (2023) Plant Disease Detection Using Vgg16. International Journal of Creative Research Thoughts [online]. 11 (1) [Accessed 05 May 2023].
2. Le, K (2021), An overview of VGG16 and NiN models [ online]. Available from: [An overview of VGG16 and NiN models | by Khuyen Le | MLearning.ai | Medium](https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484) [Accessed 20th April 2023].
3. Li, Y., Nie, J. and Chao, X. (2020) ‘Do we really need deep CNN for plant diseases identification?’, *Computers and Electronics in Agriculture*, 178, p. 105803. Available at: <https://doi.org/10.1016/j.compag.2020.105803>.
4. Mangina, E. *et al.* (2022) ‘Plant Species Detection Using Image Processing and Deep Learning: A Mobile-Based Application’, in D.D. Bochtis et al. (eds) *Information and Communication Technologies for Agriculture—Theme II: Data*. Cham: Springer International Publishing (Springer Optimization and Its Applications), pp. 103–130. Available at: <https://doi.org/10.1007/978-3-030-84148-5_5>.
5. Plant Village Disease Classification | Acc: 99.6% (2023) [online]. Available from: [Plant Village Disease Classification | Acc: 99.6% | Kaggle](https://www.kaggle.com/code/abdallahwagih/plant-village-disease-classification-acc-99-6) [Accessed 5th May 2023].
6. PlantVillage Dataset (2019) [online]. Available from: [PlantVillage Dataset | Kaggle](https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset) [Accessed 18th April 2023].