# XAI Project Report
# Dataset 4 - Group 2

***Consoloni Marco, Dalla Noce Niko, Genovese Donatella***

Explainable AI.

marco.consoloni@phd.unipi.it, niko.dallanoce@phd.unipi.it, donatella.genovese@uniroma1.it.

Date: 02/03/2024

`https://github.com/nikodallanoce/XAI_PhD_prj`

# Contents

# List of Figures

# 1 Introduction

This document provides the report of the project conducted for the XAI Project 2023-2024. This project aims at training a machine learning model for plant disease classification task and exploring it by using Explainable Artificial Intelligence (XAI) techniques.

The structure of the document is outlined as follows. Section 2 delineates the contextual background of the project, outlining primary requisites for real-world applications of plant disease detection using computer vision. Section 3 discusses the details of the dataset used for training the model, the model architecture, the training process of the model, and the training process of an autoencoder for the ABELE method. In Section 4, we discuss the results derived from employing XAI techniques on our model, presenting the analysis of six distinct case studies. Finally, in Section 5 we outline the principal constraints encountered in our project, while also discussing potential future studies that could prove beneficial for real-world applications of XAI techniques in the filed of plant disease detection.

# 2 Plant Disease Detection Domain

Plant diseases and pests detection is a very important research content in the field of machine vision. It is a technology that uses machine vision equipment to acquire images to judge whether there are diseases and pests in the collected plant images. At present, machine vision-based plant diseases and pests detection equipment has been initially applied in agriculture and has replaced the traditional naked eye identification to some extent [16]. According to [11], the plant disease and pest detection task for machine vision can be broken down in three different levels: “what”, “where” and “how”. These levels are represented in Figure 2.0.1. The ”what” stage involves classification akin to computer vision, predicting the categorical information of images. In the subsequent ”where” stage, the focus shifts to localization, identifying the diseased or infested areas within images through bounding boxes. This stage not only discerns the types of diseases and pests present but also pinpoints their exact locations. Finally, the ”how” stage corresponds to segmentation, aiming to meticulously separate unhealthy tissues or pest agents from the background on a pixel-by-pixel basis [16].

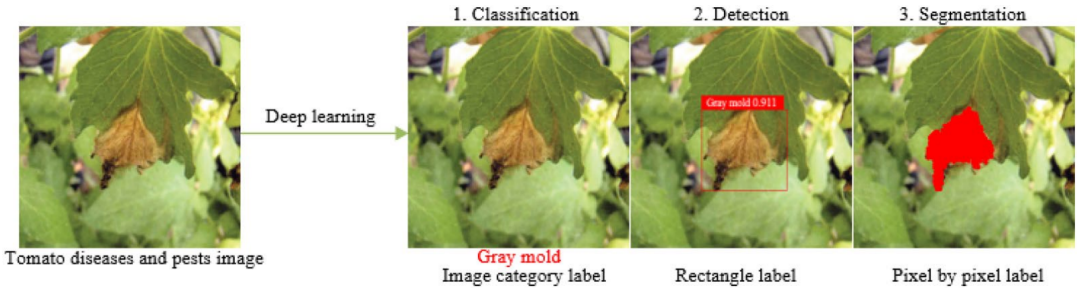



Figure 2.0.1: The three stages of Plant Disease and Pest Detection for computer vision. Adapted from [16]

This project focuses only on plant disease detection. From the background literature, we identified the primary requirements for real-world applications of plant disease detection as follows:

- Early detection of plant diseases during initial phases of infection, characterized by symptoms often imperceptible to the majority of human observers [17][18].
- Tracking the advancement of plant disease for the application of appropriate management strategies. [18][16][13]
- Designing compact deep learning models for disease detection, suitable for on-field crop disease diagnosis via smartphone technology."[17].
- Designing systems for practical deployment in real-world scenarios where images may exhibit non-homogenous backgrounds (i.e., total black or total white background). AI models should be capable of accurately classifying disease manifestations directly on the plant, regardless of background complexity [2][15][13].
- Designing systems capable of effectively detecting diseases both on single and multiple overlapping leaves, which is crucial due to the diverse conditions encountered in real-world scenarios [13].

These points will serve as guidelines for developing the case studies in section 4.

# 3 Setup

## 3.1 Dataset

In this paragraph, we describe the dataset employed for the machine learning task, furnishing the principal metadata. The dataset is called "New plant diseases"[1] and it has been created using offline augmentation from the original "Plant Village" dataset[2]. The dataset contains approximately 87,000 RGB images of both healthy and diseased crop leaves from 14 different plant species, classified into 38 distinct categories. Within them, 26 are leaves with diseases and the rest are healthy. This dataset comprises a balanced number of samples for each class and has already been divided into training, validation, and test sets. The training set consists of approximately 70K instances, the validation set of roughly 17K, and the test set of only 33. We report the main characteristics of the dataset we used in Figure 3.1.1 and the number of samples of each leaf in Table A1.

*Plant Names*

| Plant Name |
|---|
| Corn_(maize) |
| Potato |
| Cherry_(including_sour) |
| Tomato |
| Pepper,_bell |
| Grape |
| Peach |
| Apple |
| Orange |
| Strawberry |
| Soybean |
| Squash |
| Raspberry |
| Blueberry |

*Dataset Characteristics*

| Characteristic | Value |
|---|---|
| Total Classes | 38 |
| Unique Plants | 14 |
| Number of Diseases | 26 |
| Total Images | 87900 |
| Number of Images (Training) | 70295 |
| Number of Images (Validation) | 17572 |
| Number of Images (Test) | 33 |
| Image Type | JPEG |
| Image Size | (256, 256) |
| Number of Color Channels | 3 |

*Class Names and Labels*

| Class Name | Label |
|---|---|
| Apple___Apple_scab | 0 |
| Apple___Black_rot | 1 |
| Apple___Cedar_apple_rust | 2 |
| Apple___healthy | 3 |
| Blueberry___healthy | 4 |
| Cherry_(including_sour)___Powdery_mildew | 5 |
| Cherry_(including_sour)___healthy | 6 |
| Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot | 7 |
| Corn_(maize)___Common_rust_ | 8 |
| Corn_(maize)___Northern_Leaf_Blight | 9 |
| Corn_(maize)___healthy | 10 |
| Grape___Black_rot | 11 |
| Grape___Esca_(Black_Measles) | 12 |
| Grape___Leaf_blight_(Isariopsis_Leaf_Spot) | 13 |
| Grape___healthy | 14 |
| Orange___Haunglongbing_(Citrus_greening) | 15 |
| Peach___Bacterial_spot | 16 |
| Peach___healthy | 17 |
| Pepper,_bell___Bacterial_spot | 18 |
| Pepper,_bell___healthy | 19 |
| Potato___Early_blight | 20 |
| Potato___Late_blight | 21 |
| Potato___healthy | 22 |
| Raspberry___healthy | 23 |
| Soybean___healthy | 24 |
| Squash___Powdery_mildew | 25 |
| Strawberry___Leaf_scorch | 26 |
| Strawberry___healthy | 27 |
| Tomato___Bacterial_spot | 28 |
| Tomato___Early_blight | 29 |
| Tomato___Late_blight | 30 |
| Tomato___Leaf_Mold | 31 |
| Tomato___Septoria_leaf_spot | 32 |
| Tomato___Spider_mites Two-spotted_spider_mite | 33 |
| Tomato___Target_Spot | 34 |
| Tomato___Tomato_Yellow_Leaf_Curl_Virus | 35 |
| Tomato___Tomato_mosaic_virus | 36 |
| Tomato___healthy | 37 |

Figure 3.1.1: **Dataset Summary:** The image shows 3 different tables which summarise the characteristics of the Dataset. **Plants Names** contains the name of plants in the dataset, **Dataset Characteristics** contains the relevant properties of the data and **Class Name and Labels** contains the names of the classes and the associated labels

[1] https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

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

## 3.2 Model for image classification

The objective of the task is to match each leaf with its corresponding plant and determine whether it is diseased. If it is, the corresponding disease should also be identified. We chose EfficientNet [14] to fulfil this task. At the time of writing, this model is one of the most powerful CNN for image classification. It is available in eight different architectures ranging from "B0" to "B7", from the smallest to the largest. Each variant represents a different trade-off between model size and accuracy, enabling users to select the appropriate model variant based on their specific requirements. The architecture we selected is B0, which is the smallest yet powerful enough to accomplish our classification task. This model has 4.1 million of parameters and takes as input a $256 \times 256$ RGB image and classifies it into one of the 38 classes. Another reason for using this model is the availability of pre-trained weights, which reduces the total training time and can improve performance. The general architecture of EfficientNet can be seen in Figure 3.2.1.

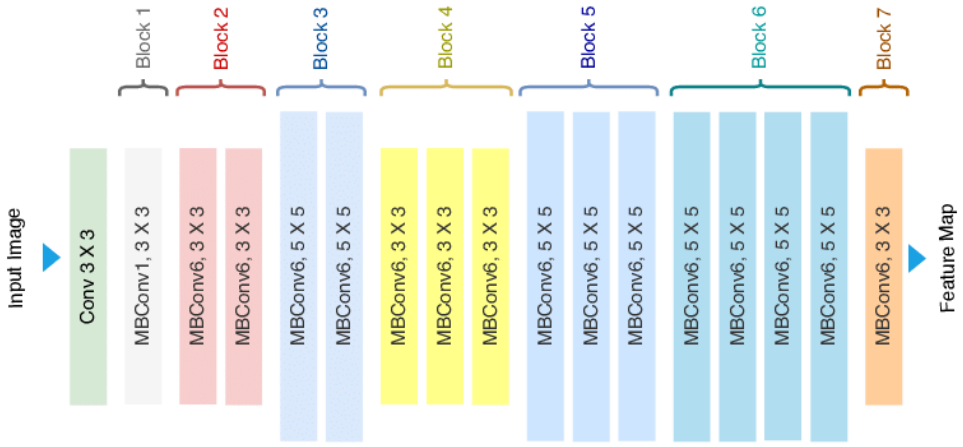



Figure 3.2.1: EfficientNet architecture.

**Training.** Training phase was performed using PyTorch Lightning [3] on four NVIDIA V100 GPUs, each with 32 GB of VRAM. A training batch consisted of 256 images and their corresponding labels. We used Adam optimizer [1] with a learning rate of $8 \times 10^{-4}$ and cosine learning rate scheduler [5]. One model was trained from scratch, while the other using pre-trained weights. The former reached convergence after 50 epochs, while the latter achieved it after only 10. Both the models use cross entropy loss with a 0.1 probability of label smoothing [4].

[3] `https://lightning.ai/`

## 3.3 Autoencoder model for ABELE

We developed a simple autoencoder for the ABELE explainer [12]. It consists of four convolutional layers on both the encoder and the decoder modules. We denote the latent space as the matrix $LT \in \mathbb{R}^{64\times32\times32}$. This dimension was chosen as a trade-off between the quality of the reconstructed images and the size of the model. With this setting, the model has 131 thousand of parameters.

**Training.** This model was trained using the same hardware described in section 3.2. We used a batch composed of 256 images, Adam optimizer with a learning rate equal to $2 \times 10^{-3}$, a cosine learning scheduler and mean squared error (MSE) loss. Training stabilized after 30 epochs.

# 4 Results

In this section, we present the results obtained through the application of XAI methodologies across a series of small case studies. Each case study encompasses multiple images, enabling comparative assessments between instances and offering a comprehensive characterization of the model's performance. For every case study, we provide a description outlining its significance within the Plant Disease Detection domain. Additionally, we specify the target users who stand to benefit from the explanations provided.

## 4.1 Assessing the image classification model performance

We asses the performance of our model by using the accuracy metric. The dataset we considered was already divided into train, validation and test set. Thus, we used a simple hold-out validation strategy. The model trained from scratch achieved 99.65 percent accuracy on the validation set. However, the one fine-tuned with pre-trained weights performed even better, achieving 99.94 percent accuracy. Both models correctly classify every instance in the test. For the purpose of the Explainable AI course, we chose to conduct all the experiments using the less accurate model. This decision gave us a larger set of misclassified images to run the experiments on. In Figure 4.1.1, we show the confusion matrix on the validation set instances. We can notice that the highest number of misclassified images occurs with class 7, "Corn (maize) Cercospora leaf spot Gray leaf spot", which is wrongly predicted as class 9, "Corn (maize) Northern Leaf Blight". To improve readability, we do not write the name of the classes in the axis. A mapping from the class index to its corresponding name can be viewed in Figure 3.1.1.

## 4.2 Autoencoder model for ABELE

In subsection 3.3 we have described the structure of the autoencoder model used for conducting the experiments using ABELE. On validation set, we got a reconstruction loss of $3.1 \times 10^{-3}$. The off-the-shelf autoencoders provided with this course were not used as they produced excessively blurred images. Even if the reconstructed image seems good, we could not make ABELE work with this model. More specifically, this method was not able to generate a prototype related to the input image to be explained. In addition, the counterfactuals produced were of the same class as the image to be interpreted. We report an example of this behaviour in Figure B.2. We also tried to let ABELE work by increasing the dimension of the latent space to $128 \times 32 \times 32$ without success. From the Figure B.1, we can notice that a larger latent space helps to improve the quality of the reconstructed image. We suspect that the reasons for ABELE's failure are not related to the quality of the reconstructed images.

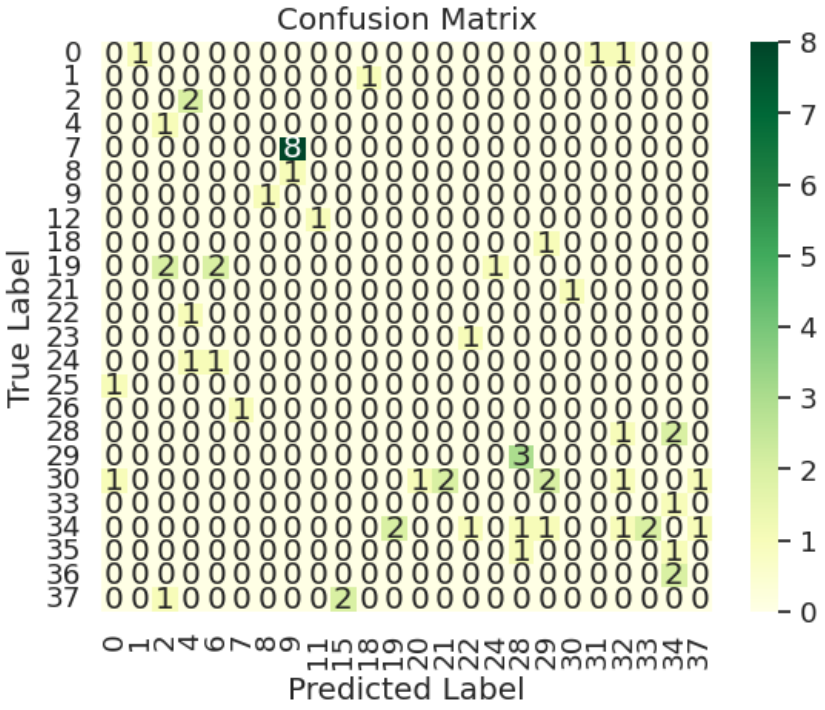



Figure 4.1.1: **Confusion matrix**: The image shows the confusion matrix of columns and rows that have misclassified instances.

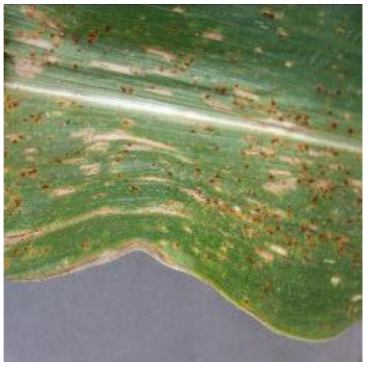
(a) Original image. (b) Reconstructed image

Figure 4.2.1: A comparison between an original image from the dataset with the image reconstructed by our autoencoder.

## 4.3 Case study 1: Misclassified Images

We commence our analysis by examining the confusion matrix of misclassified images depicted in Figure 4.1.1. The matrix reveals a notable observation: there are 8 instances of class 7 (Corn(maize) Cercospora leaf spot) misclassified as class 9 (Corn(maize) Northern Leaf Blight). This discrepancy hints at potential challenges the model faces in accurately distinguishing certain images belonging to class 7. Therefore, our subsequent endeavor is to leverage explainability techniques to gain insights into the elements within these images contributing to the misclassification. To this end, we consider (as illustrated in Figure 4.3.1): an instance of class 7 misclassified as class 9, a representative of class 7 and representative of class 9.

(a) **Label:** Corn(maize) Cercospora leaf spot(7); **Prediction:** Corn(maize) Northern Leaf Blight(9); **Confidence:** 0.91



(b) **Label:** Corn(maize) Cercospora leaf spot(7); **Prediction:** Corn(maize) Cercospora leaf spot(7); **Confidence:** 0.91

(c) **Label:** Northern Leaf Blight(9); **Prediction:** Northern Leaf Blight(9); **Confidence:** 0.91

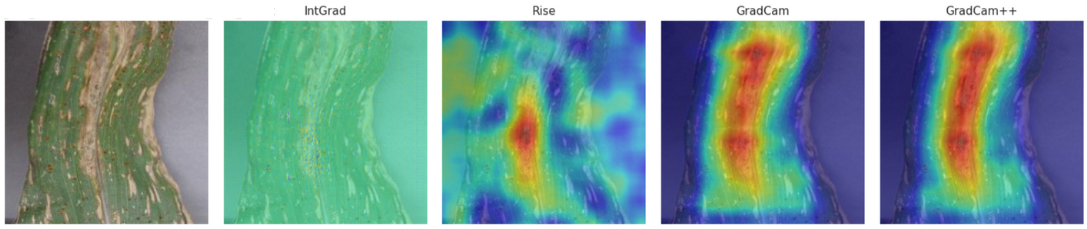
Figure 4.3.1: **Images to analyze**: The image shows the examples that we will analyze with explainable techniques.

As can be seen from Figure 4.3.1 Cercospora Leaf Spot is characterized by circular lesions in its initial stages, marked by small, tan to grayish spots bordered by dark brown to purple hues. Over time, these spots may merge, forming irregular shapes. In contrast, Northern Leaf Blight exhibits cigar-shaped or elliptical lesions, initially appearing as small, grayish-green spots with tan centers. These lesions often grow larger and elongated, frequently aligning with leaf veins. Hence we proceed by implementing the following Explainability techniques: Integradient Gradient, Rise, GradCam and GradCam ++ in order to detect which characteristics of Northern Leaf Blight are present in the misclassified instance. In examining the XAI methods (see Figure 4.3.2) applied to our analysis, distinct patterns emerge for images belonging to Class 7 and Class 9. For images representing Class 9, all XAI techniques consistently highlight the presence of cigar-shaped lesions, a characteristic trait of this class. This consistency suggests that

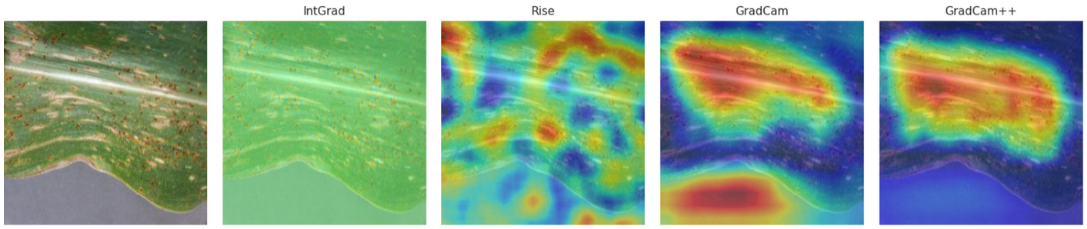
the model heavily relies on identifying these lesions for classification purposes. However, when analyzing images from Class 7, the XAI outputs reveal diverse focal points across the image, indicating a lack of clear discriminative features for Class 7 classification. Interestingly, for the misclassified image, XAI techniques converge on the center of the image, similar to the cigar-shaped lesion characteristic of Class 9. This convergence suggests a potential misinterpretation by the model, which is further supported by deletion metrics (see Figure 4.3.3) showcasing rapid shifts in accuracy upon pixel manipulation within this region. The previous analysis suggests also some further experiments that can be conducted from developers, such as a deep investigation of discriminative features for class 7. This could involve the analysis of different images from that class to uncover common patterns reflected in heat maps. It could also be useful to compare these analysis combining various tasks such as plant-specific classification or distinguishing healthy from non-healthy plants.

The last XAI method we employed for this type of analysis is SHAP [6]. This method highlights in red the parts of the image that contributes positively to the classification for a certain class and in blue the part that would shift the classification to another class. Given an image, this method allows us to explain both a specific layer of the whole network. For the first case, we used the gradient explainer. In Figure 4.3.4 we can see, from right to left, the misclassified image to be explained and the most likely predictions, sorted by the probability of being classified with other labels. We can get the most important results from the whole network explanation. Apparently, the model tends to focus on the central part of the image in order to classify it poorly as class 9. This result is similar to that obtained by RISE, which showed that the models tend to focus on the same part to make the prediction. The second most likely image is the correct class, while the third is the one with the lowest probability of being correctly classified. The probability is so low that we cannot even distinguish the red sections from the blue ones.

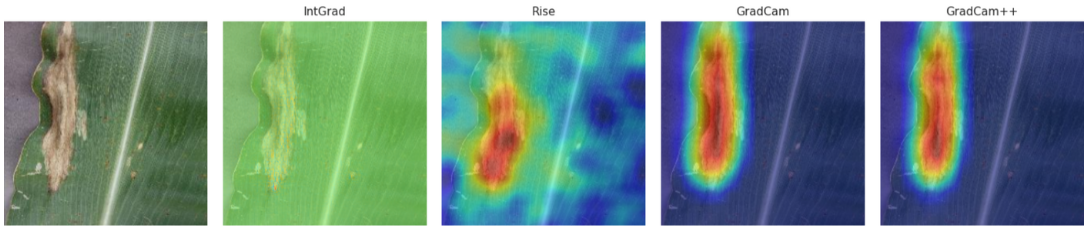
Finally, we note that all the explanations we have given could also be effective for farmers, since it could be useful to know that for class 7 the algorithm should misinterpret an image as class 9 due to some apparently similar characteristics present in the two diseases.




(a) Results of explainability techniques Cercorspora Leaf Spot (7) misclassified as Northern Leaf Blight (9)

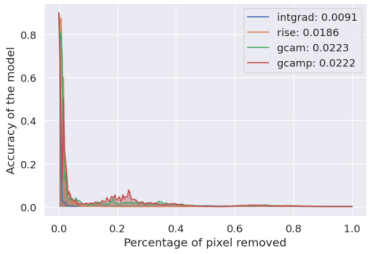



(b) Results of explainability techniques Cercorspora Leaf Spot (7) correctly classified

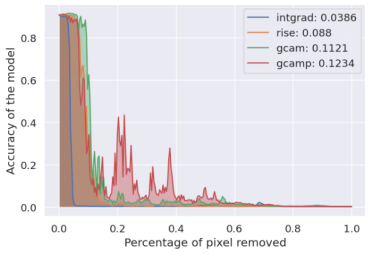



(c) Results of explainability techniques Northern Leaf Blight (9) correclty classified

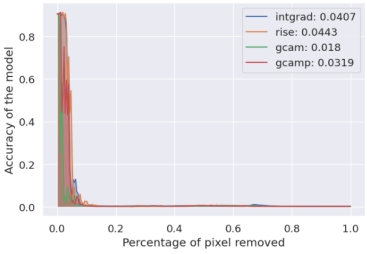
Figure 4.3.2: **Explainability results**: The image shows the explainability results of the instances with IntGrad, Rise, GradCam and GradCam++.




(a) Deletion Metric for Cercorspora Leaf Spot (7) misclassified as Northern Leaf Blight (9)

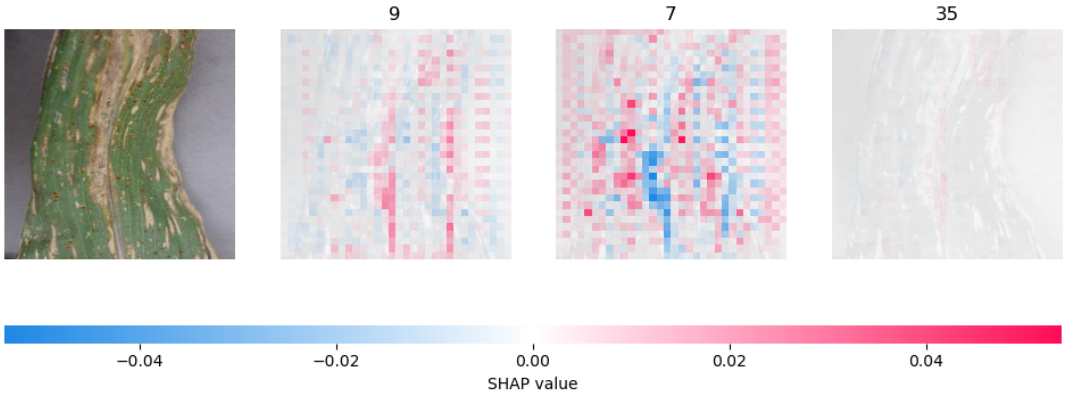



(b) Deletion Metric for Cercorspora Leaf Spot (7) correctly classified

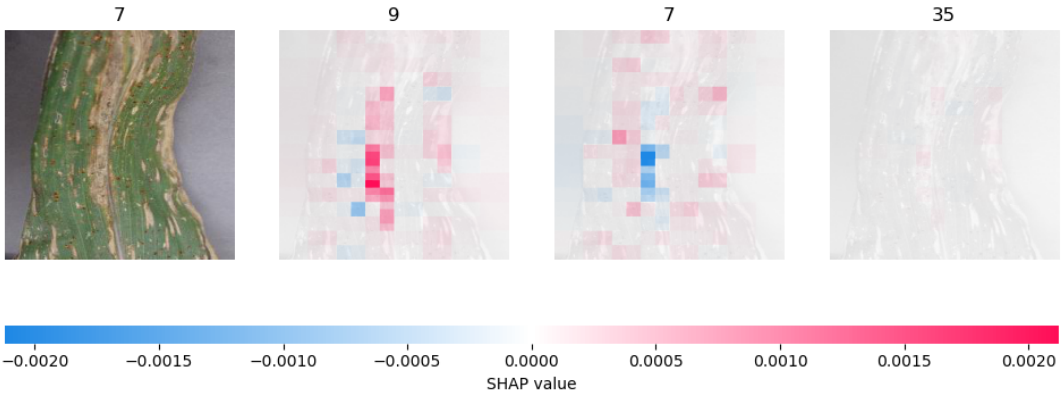



(c) Deletion Metric for Northern Leaf Blight (9 correclty classified)

Figure 4.3.3: **Deletion Metric**: The image shows the deletion metric of different explainability methods.




(a) Output of the gradient explainer for the 5-th convolutional layer.




(b) Output of SHAP taking into account the entire network.

Figure 4.3.4: A comparison between the explanations produced by SHAP. In Figure (a) we consider only the 5-th layer. Instead, in Figure (b), the whole network.

## 4.4 Case study 2: Non-Homogeneous Background

In the field of machine vision for plant disease detection, it is of primary importance designing systems for practical deployment in real-world settings where images may exhibit non-homogenous backgrounds such as those with total black or total white backgrounds. The objective of this case study is to assess if our model is capable of accurately classifying disease symptoms directly on the plant, regardless of background complexity. To this end, we have specifically chosen four images belonging to the class "Strawberry Leaf scorch (26)", which were captured in field conditions and exhibit non-uniform backgrounds. These four images have been correctly classified by our model. Notably, the class "Strawberry Leaf scorch (26)", is unique among the dataset classes in presenting such non-homogeneous background. Given this distinctive characteristic of this class, we investigated how the complexity of the background in these images influences the model's

predictions. As shown in Figure 4.4.1 the disease traits associated with “Strawberry Leaf scorch (26)” are as follows: leaf scorch symptoms manifest as numerous small, irregular, red/purplish spots or ”blotches” that develop on the upper surface of leaves.

Figure 4.4.1: Examples of leaf scorch disease on strawberry leaves.

We applied the following XAI attribution methods to our model: IntGrad, Rise, Lime GradCam and GradCam++, in order to generate saliency maps for each of the four images (instances) featuring non-uniform backgrounds. Figure 4.4.2 shows the superimposition of the four original images with the saliency maps generated by the attribution methods.

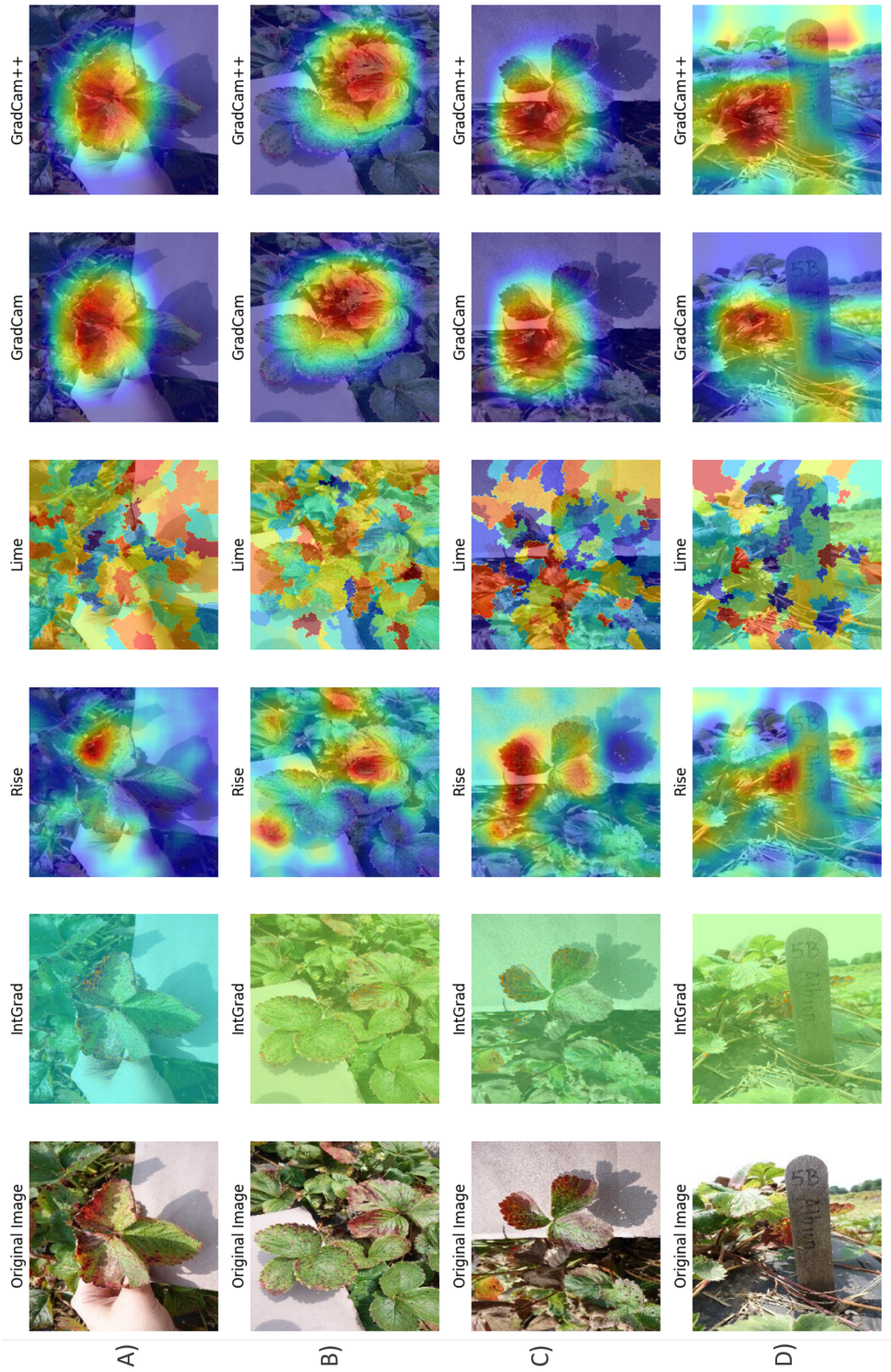



Figure 4.4.2: Saliency maps of four images of strawberry leaves affected by the leaf scorch disease with non-homogeneous background.

From Figure 4.4.2, it is evident that the Lime method fails to capture distinctive features of the leaf scorch disease. The high-valued superpixels identified by Lime do not correspond to the disease symptoms. Conversely, when examining images A and B, both Intagrad and Rise methods appropriately focus on the relevant areas of the images, providing accurate explanations for the location of the disease symptoms. Moreover, it seems that the methods are able to capture the symptoms of the disease's not only on the leaf on the foreground but also on the leaves on the background of the image. This observation warrants further investigation, especially considering that our model was trained on images with normalized backgrounds.

Upon closer examination of image C, IntGrad, Rise, GradCam and GradCam++ methods focus not only on the right part of the image but also on red/purplish spots of the background. These red/purplish spots do not indicate leaf scorch symptoms, but rather the normal coloration of other leaves in the background.

Furthermore, an examination of image D reveals some other limitations of our model. Image D does not depict a single leaf; it appears to be a wooden stick commonly used in agriculture by farmers to indicate cultivation names. IntGrad, Rise, GradCam and GradCam++ methods focus on red/purplish areas of the images which do not represent leaf scorch symptoms. Indeed, according to the saliency maps, the model appears to concentrate on a group of stems in the foreground of the image that exhibit a red/purplish coloration. However, it is important to note that red pigment in stems is normal for strawberry plants and does not indicate leaf scorch disease.

It is to be considering, that the leaf scorch disease is the only disease in the entire dataset that manifests with red/purplish spots. As a result, our model may correctly predict the leaf scorch disease in Strawberries by simply correlating red/purplish spots (i.e., colour) with the leaf scorch disease (i.e., class), without having a deep characterization of the symptoms of the disease. This hypothesis is supported by the misleading explanations provided by attribution methods for images C and D, where the model correctly classifies the images but focuses on meaningless areas for the prediction of the disease.

To have a deeper understanding into the issue of "red/purplish" colour and leaf scorch disease correlation, we conducted an additional test as outlined below: 1) we choose an image of a healthy leaf accurately classified by the model with a high output probability (i.e., high confidence); 2) we randomly alter some pixels in the image to red and 3) we assess whether the presence of the red pixels influences our model's prediction, potentially causing it to misclassify the healthy image as strawberry leaf scorch.

We decided to carry out the experiment on an image belonging to the "Grape healthy (14)" class from the validation set, as the characteristics of grape plants are notably distinct from those of strawberry plants and, moreover, grape plants are immune to leaf scorch disease.

The outcomes of the test are presented in Figure 4.4.3, which displays two images: image a) is the original image, while image b) shows the perturbated image where we have

altered 600 pixels of the 256x256 image (0.09% perturbation). Each image is accompanied by the class label (Label), the predicted class (Prediction) and the output probability (Confidence).

(a) Original image. **Label:** Grape healthy(14); **Prediction:** Grape healthy(14); **Confidence:** 0.932

(b) Perturbated image. **Label:** Grape healthy(14); **Prediction:** Strawberry Leaf scorch(26); **Confidence:** 0.896

Figure 4.4.3: Image perturbation with red color pixels of a grape healthy leaf.

Figure 4.4.3 clearly shows that the model misclassified the perturbed image with a confidence of 0.896 as a strawberry leaf scorch, indicating that the model has merely learned a correlation between "red/purplish" colour and leaf scorch disease.

To sum up, despite the high accuracy of the model in the classification task, the XAI techniques have highlighted potential limitation of our model in understating the specific characteristics of the leaf scorch disease, as well as possible challenges in dealing with images with non-normalize backgrounds.

## 4.5 Case study 3: Low Confidence Predictions

In this case study, we conducted an analysis of images that were correctly classified by our model but with a low output probability for the predicted class, indicating a low confidence in the model's prediction. The objective of this investigation is to explain why our model demonstrates low confidence for certain predictions and which features of the images contribute to this lack of confidence. To this end, we have found that 55 images out of the 17,572 of the validation set are classified with an output probability less than 0.7. Among these 55 images, we discovered that those with an excessively high level of brightness corresponded to the lowest prediction probabilities. We selected the two images with the lowest confidence values. Figure 4.5.1 showcases the two images,

each accompanied by the class label (Label), the predicted class (Prediction) and the output probability (Confidence). These images exhibit a consistent visual pattern where parts of the images are clearly distinguishable, while other parts appear entirely white. It is likely that these images were generated using data augmentation techniques, as the illuminance conditions do not resemble those typically found in real-world images.

(a) **Label:** Blueberry healthy(4); **Prediction:** Blueberry healthy(4); **Confidence:** 0.401

(b) **Label:** Tomato healthy(37); **Prediction:** Tomato healthy(37); **Confidence:** 0.484

Figure 4.5.1: Examples of images predicted by the model with low output probability and with excessively high level of brightness.

We applied XAI attribution methods to our model to generate saliency maps. Figure 4.5.2 showcases the superimposition of the original images with the saliency maps generated by the attribution methods.

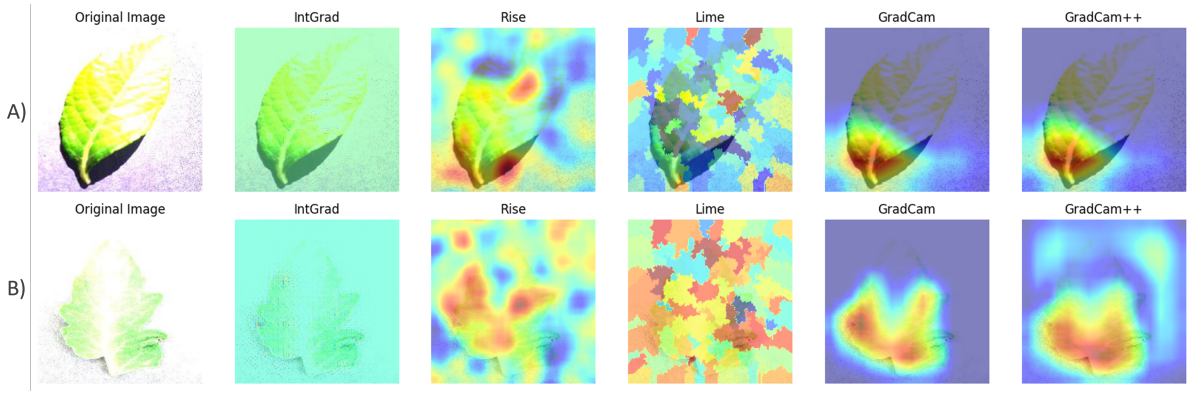



Figure 4.5.2: Saliency maps of two images with excessively high level of brightness.

From Figure 4.5.2, it is evident that the Lime method fails to capture distinctive features of the leaves under analysis. The high-valued super pixels identified by Lime do not correspond to distinguishable parts of the leaves; instead, they align with the white/meaningless areas of the images. On the other hand, IntGrad, Rise, GradCam, and GradCam++ demonstrate that the distinguishable and coloured parts of the leaves contribute positively to the model's prediction. The explanations provided for these images allow us to conclude that the model's prediction seems to be sound even under these extreme illuminance conditions. This case study can benefit developers of machine vision systems for plant disease detection by guiding them in tuning data augmentation techniques to generate images that more closely resemble real-world illuminance conditions. Moreover, it can provide guidance for end-users in classifying plants that are not well recognizable due to the low image quality.

## 4.6 Case study 4: High Confidence Predictions

In this case study, we conducted an analysis of images of healthy leaves that were correctly classified by our model with a high output probability for the predicted class, indicating a high confidence in the model's prediction. Given the high confidence of predictions, the objective of this investigation is to analyse if the model attends to the distinctive visual traits of healthy plants. In other words, we aim to determine if the model explains the instances based on the distinctive aspects of plant species, regardless of the presence of diseases. To this end, we chose to conduct this analysis on the images belonging to the "Grape healthy (14)" class. Grape leaves are known for their heart-shaped appearance with multiple lobes. They typically grow in an alternate pattern and have serrated, or toothed, edges with pointed tips on each lobe. These unique traits visually distinguish grape leaves from all other healthy leaves in the entire dataset, even for individuals without expertise in plant identification, making them particularly suitable for this case study.

We choose the two images of the "Grape healthy (14)" class from the validation set with the highest output probability values. Figure 4.6.1 shows the two images, each accompanied by the class label (Label), the predicted class (Prediction) and the output probability (Confidence).

We applied IntGrad, GradCam and GradCam++ attribution methods to our model to generate saliency maps. Figure 4.6.2 illustrates the superimposition of the original images with the saliency maps generated by the attribution methods.

(a) **Label:** Grape healthy(14); **Prediction:** Grape healthy(14); **Confidence:** 0.932

(b) **Label:** Grape healthy(14); **Prediction:** Grape healthy(14); **Confidence:** 0.935

Figure 4.6.1: Examples of two images of "Grape healthy (14)" class predicted by the model with high output probability.

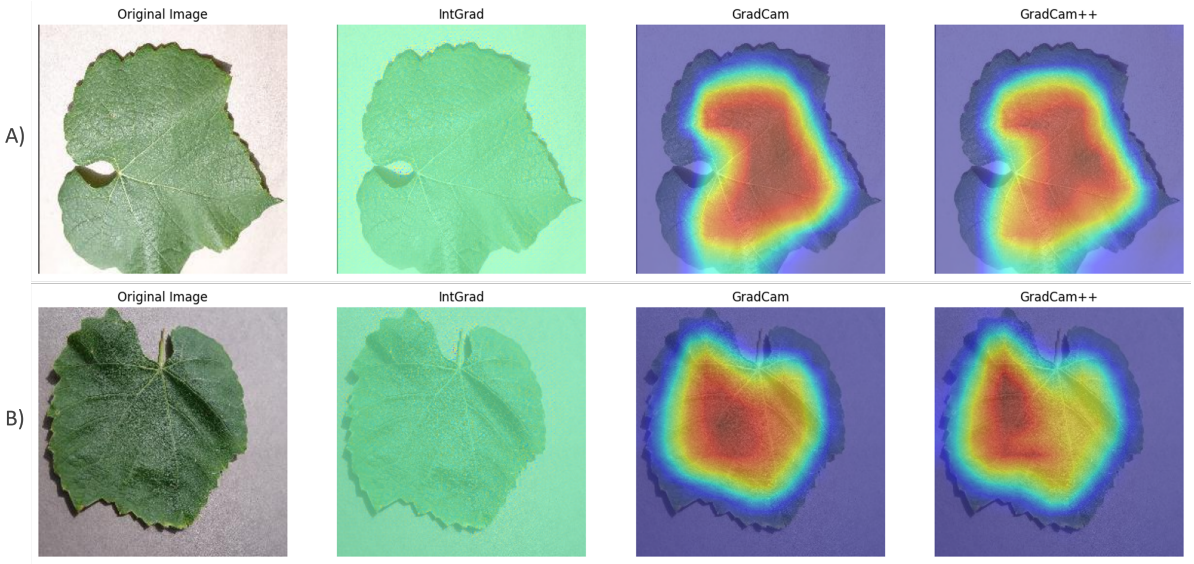



Figure 4.6.2: Saliency maps of two images of "Grape healthy (14)" class predicted by the model with high output probability.

The analysis of Figure 4.6.2 reveals that IntGrad primarily highlights the edges of the images, delineating the serrated contours of the leaf perimeters. Conversely, GradCam and GradCam++ emphasize the entirety of the surface of the leaves, resulting in saliency

maps with a distinctive heart-shaped pattern with multiple lobes. Integrating these analyses provides a comprehensive characterization of the features of healthy grape leaves.

## 4.7 Case study 5: Similar Images

In this case study, we performed a comparative analysis of visually similar images of unhealthy leaves belonging to the same class. The objective of this case study is to assess the stability of attribution methods across two explanations, expecting similar instances to receive comparable explanations. To this end, we manually selected two visually similar images of the "Potato Early blight (20)" class from the validation set, as depicted in Figure 4.7.1. Notably, we evaluated these two images to be visually similar due to their common spatial orientation, leaf coloration, illuminance condition and infection stage.

(a) **Label:** Potato Early blight(20); **Prediction:** Potato Early blight(20); **Confidence:** 0.920

(b) **Label:** Potato Early blight(20); **Prediction:** Potato Early blight(20); **Confidence:** 0.908

Figure 4.7.1: Examples of two visually similar images of "Potato Early blight (20)" class.

We applied XAI attribution methods to our model to generate saliency maps. Figure 4.7.2 showcases the superimposition of the original images with the saliency maps generated by the attribution methods.

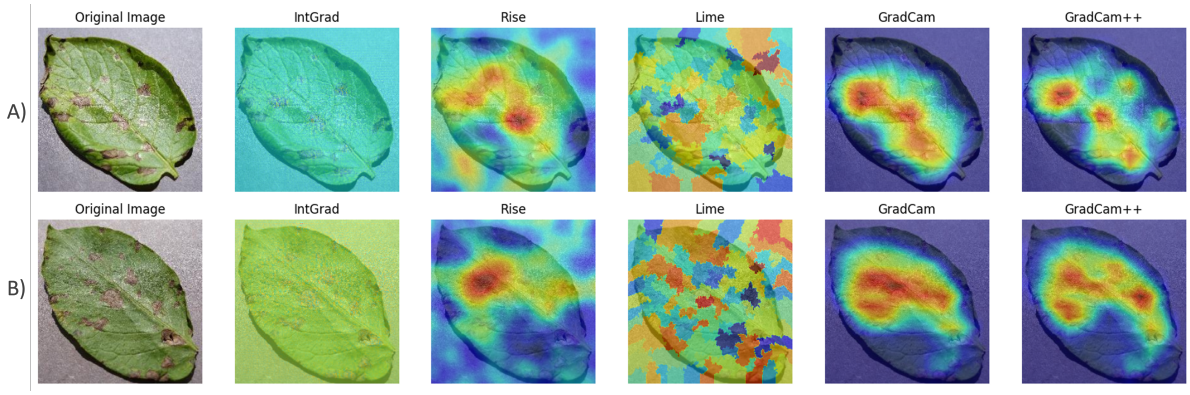



Figure 4.7.2: Saliency maps of two visually similar images of "Potato Early blight (20)"class.

The examination of Figure 4.7.2 indicates that the explanations provided by each attribution method for the two images are highly similar and overlapping, implying consistency in the model's classification of early blight disease for visually similar instances.

However, these explanations do not adequately characterize the early blight disease affecting potato plants. Early blight disease manifests in potato plants as circular to angular dark brown lesions 0.12 to 0.16 inch (3–4 mm) in diameter. Upon closer examination of Rise, GradCam and GradCam++ saliency maps, we note that our model predominantly focuses on the unhealthy areas of the leaf surfaces to drive class predictions. Nonetheless, it primarily attends to the major lesions on the leaves, failing to detect a substantial portion of the overall unhealthy tissue of the leaves.

These shortcomings restrict the potential real-world applications of the explanations. While the saliency maps' explanations are beneficial for classification tasks providing the major visual indicators/pointers to unhealthy tissue on leaves, they are insufficient for accurately locating all lesions within images. This deficiency poses challenges for segmentation tasks, which aim to separate lesions from the background pixel by pixel, thereby impeding higher-level severity evaluations of plant diseases.

## 4.8 Case study 6: Multi-Leafed Plant

In this case study we investigate some images from class 37 (Tomato Healthy) which present multiple leaves (see Figure 4.8.1). Since our current dataset is mainly composed by a single leaf while real-world application should ideally be capable of classifying images containing more then one leaf simultaneously, it is useful to understand if our model is able to explain multi-leafed images.

As shown in Figure 4.8.3 we have used 5 different explainability techniques: IntGrad, Rise, Lime, GradCam and GradCam++. We can see that IntGrad and Lime seem to not highlight the discriminative parts of the image for all cases (a,b and c) as it is confirmed

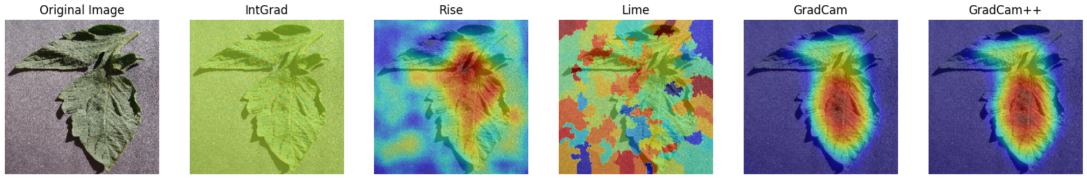
(a) **Label:** Tomato healthy(37); **Prediction:** Tomato healthy(37); **Confidence:** 0.912

(b) **Label:** Tomato healthy(37); **Prediction:** Tomato healthy(37); **Confidence:** 0.935

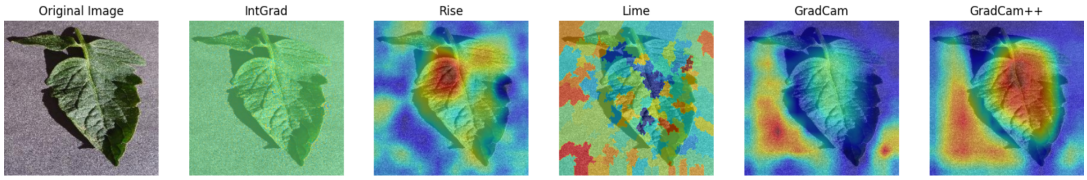
(c) **Label:** Tomato healthy(37); **Prediction:** Tomato healthy(37); **Confidence:** 0.935

Figure 4.8.1: **Images with multiple leaves**: The image shows 3 instances from the class 37 (Tomato Healthy) which present multiple leaves.

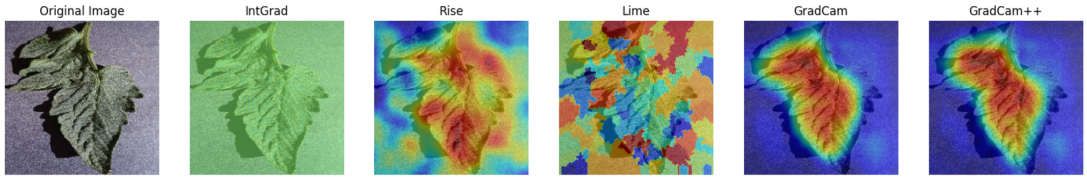
also from Deletion metric where we can see that the accuracy is reduced only when about 70% of pixels are removed, which suggest that the pixels with more relevance are not the most important for the accuracy. As regards the other methods for case a) RISE, GradCam and GradCam++ highlight only one leaves as representative for the prediction, for case b) all the remaining methods do not seem to highlight the two leaves. and for case c) RISE seems not to detect well the leaves while GradCam and GradCam++ correclty highlight the 2 leaves. Hence, this analysis suggests that in some cases the model may ignore one of the two leaves, especially in cases where they are well separated (as in a) and b), while it seems to better recognize the two parts where they are not well separated (as in c)). From a developers point of view this Analysis can be helpful to understand the limitation of the model, in particular there could be the possibility that the model have some difficulties to recognize 2 leaves present in the same image, probably also due the fact that most of images in the training set present only one leaf.




(a) Explainability results for image (a)

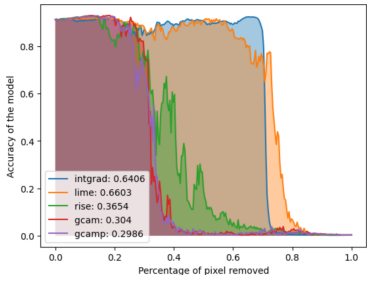



(b) Explainability results for image (b)

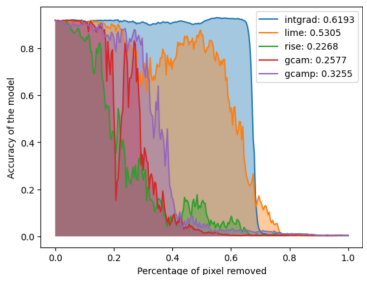



(c) Explainability results for image (c)

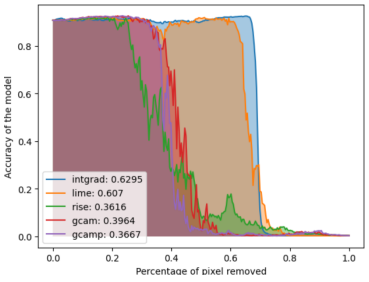
Figure 4.8.2: **Explainability results**: The image shows the explainability results with IntGrad, Rise, Lime, GradCam, GradCam++




(a) Deletion Metric image (a)




(b) Deletion Metric for image (b)




(c) Deletion Metric for image (c)

Figure 4.8.3: **Deletion Metric**: The image shows the deletion metric of different explainability methods.

# 5 Conclusions and future developments

In this section we outline the main limitations and future developments of this project.

As for the limitations, we provide explanations solely relying on saliency maps abstaining from the consideration of counterfactuals and prototypes due to the operational challenges encountered with the implementation of ABELE. Consequently, our capacity to furnish explanations was restricted to the analysis of the most pivotal pixels governing the model's decision-making at a local level. Another limitation stems from the dataset, which subsequently affects the classification task. Specifically, by using 38 classes that contain both crop species and disease status, we have made the classification task harder than ultimately necessary for real-world applications, as growers are expected to know which crops they are growing. Another limitation concerning the practical applications of the explanations provided by the XAI methods is the computational time required. Currently, it takes 1-3 minutes to compute these saliency maps using Colab GPUs. The computational cost impedes real-world applications, particularly in scenarios where farmers, while out in the fields, could benefit solely from smartphone applications for assistance [17].

For future studies, we plan to address the operational challenges encountered with the implementation of ABELE. Moreover, it would be more beneficial to construct and deploy a dataset focusing on different stages of diseases affecting plants. Including the "time" dimension in a dataset for "plant disease and pest detection" tasks would allow the development of a machine learning model capable of classifying different stages of diseases across various plant species, enabling: 1) early detection of disease and 2) tracking the advancement of plant diseases for the application of appropriate farming strategies.

# A Dataset

| Plant leaf (healty or ill) | Number of samples |
|---|---|
| Apple Apple scab | 2016 |
| Apple Black rot | 1987 |
| Apple Cedar apple rust | 1760 |
| Apple healthy | 2008 |
| Blueberry healthy | 1816 |
| Cherry (including sour) Powdery mildew | 1683 |
| Cherry (including sour) healthy | 1826 |
| Corn (maize) Cercospora leaf spot Gray leaf spot | 1642 |
| Corn (maize) Common rust | 1907 |
| Corn (maize) Northern Leaf Blight | 1908 |
| Corn (maize) healthy | 1859 |
| Grape Black rot | 1888 |
| Grape Esca (Black Measles) | 1920 |
| Grape Leaf blight (Isariopsis Leaf Spot) | 1722 |
| Grape healthy | 1692 |
| Orange Haunglongbing (Citrus greening) | 2010 |
| Peach Bacterial spot | 1838 |
| Peach healthy | 1728 |
| Pepper, bell Bacterial spot | 1913 |
| Pepper, bell healthy | 1988 |
| Potato Early blight | 1939 |
| Potato Late blight | 1939 |
| Potato healthy | 1824 |
| Raspberry healthy | 1781 |
| Soybean healthy | 2022 |
| Squash Powdery mildew | 1736 |
| Strawberry Leaf scorch | 1774 |
| Strawberry healthy | 1824 |
| Tomato Bacterial spot | 1702 |
| Tomato Early blight | 1920 |
| Tomato Late blight | 1851 |
| Tomato Leaf Mold | 1882 |
| Tomato Septoria leaf spot | 1745 |
| Tomato Spider mites Two-spotted spider mite | 1741 |
| Tomato Target Spot | 1827 |
| Tomato Tomato Yellow Leaf Curl Virus | 1961 |
| Tomato Tomato mosaic virus | 1790 |
| Tomato healthy | 1926 |

Table A1: Information on the types and quantities of different leaves.

# B Autoencoder for ABELE

(a) Original image. (b) Reconstructed image

Figure B.1: A comparison between an original image from the dataset with the image reconstructed by our autoencoder with a latent dimension of $128 \times 32 \times 32$.


Figure B.2: Counterfactuals produced by ABELE. All the generated images have been classified by the model as the image to be explained.

# C Explainable AI techniques

In this section, we provide a brief overview of the functioning of Explainable AI techniques employed in our experiments. We emphasize instances where these techniques may not yield satisfactory results and offer justifications for the parameters chosen.

## C.1 LIME

LIME (Local Interpretable Model-Agnostic Explanations) [3] is a Local Surrogate model whose aim is to find a local approximation of a black box model using white box ones (like linear models). Here we provide a summary of Lime algorithm:

1. Lime takes an image **x** and turns it in a vector of superpixels **x'** which express presence/absence.
2. it generate a neighborhood by randomly perturbing **x'**.
3. The black box model assigns a label to the neighborhood.
4. A linear model is trained using the neighborhood as data and superpixels as features.
5. The weights of the linear model serve as explanations for each superpixel.

As we have seen from our analysis, LIME fails in quite all cases we have tested, this could happen because:

- The superpixel transformation may fail to accurately capture the presence or absence of features in the image.
- The perturbation process may not effectively simulate diverse instances, potentially removing crucial information rather than introducing meaningful variations.
- Lime assumes local linearity in the black box model, which may not hold true, particularly in the case of complex models like deep neural networks such as EfficientNet.

## C.2 Integrated Gradient

Integrated Gradient [8] is an attribution model that generates a relevance score $R_i(\mathbf{x})$ for each pixel, indicating, for an image $\mathbf{x}$, the extent to which pixel $i$ contributes to explaining the model's classification decision. The general formula is given by:

$$R_i(\mathbf{x}) = (x_i - \tilde{x_i}) \int_0^1 \nabla f(\tilde{\mathbf{x}} + t(\mathbf{x} - \tilde{\mathbf{x}}))dt,$$

where $\mathbf{x}$ is the image we want to explain and $\tilde{\mathbf{x}}$ is a baseline image (e.g. a white or black image).

For practical computation, we can follow these steps:

1. Define a baseline and interpolate a series of images between the baseline and the current image.
2. Calculate the gradients.
3. Integrate the gradients using interpolation techniques, such as the Riemann trapezoid method.

In our experiments, Integrated Gradient seems to underperform compared to other models. This could be due to several reasons:

- It heavily relies on baseline selection, so conducting experiments with different baselines could be beneficial.
- The gradient of deep neural networks may suffer from a phenomenon called "Shattered Gradient" (REF), where the gradient exhibits high variability and excessive noise. This could contribute to providing less clear explanations for the model's behavior.

## C.3 RISE

RISE [10] is a masking method, hence it generates perturbations or masks applied to input data to assess the importance of different regions/features for a model's predictions. Here there is a general overview on how RISE works:

1. RISE generates a set of random binary masks by randomly sampling pixels from a Bernoulli distribution and apply them on the original image to create perturbed versions of it.
2. The perturbed images are fed into the deep neural network, and the model's predictions are recorded for each perturbed image.
3. The importance of each pixel in the original image is estimated based on how much the model's predictions vary when the corresponding region is masked out.
4. The importance scores obtained for each pixel across multiple perturbed images are aggregated to obtain a final importance map.

In our experiments RISE seems to work well for some images while for others it seems not to capture the relevant regions. This could happens for different reasons:

- It depends on the quality and diversity of the random masks generated during the sampling process.
- It generates importance maps by independently perturbing different regions of the input image and it could lead to not capture spatial relationships.
- It analyzes the importance of individual pixels or regions in the input image but may overlook higher-level patterns or features that contribute to the model's decision-making process.

## C.4 GradCAM and GradCAM++

Grad-CAM (Gradient-weighted Class Activation Mapping) [7] is an interpretability technique used to visualize and understand the decision-making process of convolutional neural networks (CNNs). It highlights the regions of an input image that are important for a model's prediction by computing the gradients of the target class score with respect to the feature maps of the last convolutional layer. This allows us to identify which features the model focuses on when making a prediction. Here we provide an overview of the algorithm:

1. Computes the gradients of the target class score with respect to the feature maps of the last convolutional layer of the neural network, which represent the importance of each feature map for the target class.
2. Calculates the importance weights for each feature map by applying global average pooling on gradients.
3. Computes a weighted combination of the feature maps, where the weights are the importance scores obtained from the gradients.

The overall procedure can be summarized by:

$$R_{ij}^c = \sum_k w_k^c A_{ij}^k, \qquad w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k},$$

where "c" refers to the class being explained, $w_k^c$ are the weights, $A_{ij}^k$ are the feature maps, $Z$ is the normalization factor, and $Y^c$ is the network prediction for class $c$. Grad-CAM++ [9] aims to improve upon Grad-CAM by adding pixel-wise weighting of the gradients of the output with respect to a particular spatial position.

Grad-CAM and Grad-CAM++ are typically applied to the last convolutional layer of a neural network. This choice is motivated by the fact that the convolutional layers closer to the output capture high-level semantic information about the input image, making them more relevant for understanding the decision-making process of the model.

However, it's important to note that Grad-CAM and Grad-CAM++ can theoretically be applied to any convolutional layer in the network, depending on the specific requirements of the application or the desired level of interpretability. Applying Grad-CAM to earlier layers may provide insights into more localized and low-level features detected by the network, while applying it to later layers may capture more abstract and high-level concepts.

## C.5 SHAP

SHAP (Shapley Additive Explanations) [6] is a method to explain individual predictions, based on the game theoretically optimal Shapley values. SHAP works by attributing importance scores to each input feature, which in the case of images, are the pixels or regions. These importance scores indicate how much each pixel or region contributes to the model's decision. Once the model has been trained, SHAP values are calculated by taking into account all possible combinations of features (pixels or regions) and measuring their impact on the model's output. The SHAP values are, then, visualized to provide insights into the model's decision-making process. For images, this often involves generating heatmaps where pixels or regions with higher SHAP values are highlighted.

# Bibliography


[1] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[2] Sharada P Mohanty, David P Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection". In: *Frontiers in plant science* 7 (2016), p. 1419.

[3] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. Ed. by Balaji Krishnapuram et al. ACM, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778.

[4] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308. URL: https://doi.org/10.1109/CVPR.2016.308.

[5] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=Skq89Scxx.

[6] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 4765–4774. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[7] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74. URL: https://doi.org/10.1109/ICCV.2017.74.

[8] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by

Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3319–3328. URL: http://proceedings.mlr.press/v70/sundararajan17a.html.

[9] Aditya Chattopadhyay et al. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 2018, pp. 839–847. DOI: 10.1109/WACV.2018.00097. URL: https://doi.org/10.1109/WACV.2018.00097.

[10] Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 151. URL: http://bmvc2018.org/contents/papers/1064.pdf.

[11] Justine Boulent et al. "Convolutional neural networks for the automatic identification of plant diseases". In: *Frontiers in plant science* 10 (2019), p. 941.

[12] Riccardo Guidotti et al. "Black Box Explanation by Learning Image Exemplars in the Latent Feature Space". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*. Ed. by Ulf Brefeld et al. Vol. 11906. Lecture Notes in Computer Science. Springer, 2019, pp. 189–205. DOI: 10.1007/978-3-030-46150-8_12. URL: https://doi.org/10.1007/978-3-030-46150-8_12.

[13] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. "Plant disease detection and classification by deep learning". In: *Plants* 8.11 (2019), p. 468.

[14] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[15] Alvaro Fuentes, Sook Yoon, and Dong Sun Park. "Deep learning-based techniques for plant diseases recognition in real-field scenarios". In: *Advanced Concepts for Intelligent Vision Systems: 20th International Conference, ACIVS 2020, Auckland, New Zealand, February 10–14, 2020, Proceedings 20*. Springer. 2020, pp. 3–14.

[16] Jun Liu and Xuewei Wang. "Plant diseases and pests detection based on deep learning: a review". In: *Plant Methods* 17 (2021), pp. 1–18.

[17] Lawrence C Ngugi, Moataz Abelwahab, and Mohammed Abo-Zahhad. "Recent advances in image processing techniques for automated leaf pest and disease recognition–A review". In: *Information processing in agriculture* 8.1 (2021), pp. 27–51.

[18] Sunil S Harakannanavar et al. “Plant leaf disease detection using computer vision and machine learning algorithms”. In: *Global Transitions Proceedings* 3.1 (2022), pp. 305–310.