DVTXAI: A Novel Deep Vision Transformer with an Explainable AI-based Framework and its Application in Agriculture

Sadia Kamal
University of Maryland, Baltimore County
Parth Sharma
Jaypee University of Information Technology
P.K. Gupta
School of Computing, Department of Data Science and Engineering,Mohan Babu University
Mohammad Khubeb Siddiqui
Mohammad.Khubeb.Siddiqui@ibm.com

AI Practice, IBM
Abhijit Dutt
University of Maryland, Baltimore County
Ankush Singh
Jaypee University of Information Technology

Research Article

Keywords: AI in agriculture, Explainable AI, Plant disease detection, transformer learning, SHAP

Posted Date: August 12th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4752298/v1

Additional Declarations: No competing interests reported.

DVTXAI: A Novel Deep Vision Transformer with an Explainable AI-based Framework and its Application in Agriculture

Sadia Kamal[a], Parth Sharma[c], P.K. Gupta[d], Mohammad Khubeb Siddiqui[b,*], Ankush Singh[c], Abhijit Dutt[a]

[a]*Department of Computer Science and Engineering,*
University of Maryland Baltimore County, USA
email: sadia1402@umbc.edu; adutt@umbc.edu
[b]*Artificial Intelligence Practice, IBM Australia, 256 George Street, Sydney 2000, Australia*
email: Mohammad.Khubeb.Siddiqui@ibm.com;khubeb@ieee.org
[c]*Department of Computer Science and Engineering,*
Jaypee University of Information Technology, Waknaghat, Solan, HP, 173 234, India
email: 211156@juitsolan.in; 211106@juitsolan.in
[d]*School of Computing, Department of Data Science and Engineering,*
Mohan Babu University, Tirupati, AP, 517102, India
email: pkgupta@ieee.org

Abstract

Agriculture is one of the fundamental components of human civilization, contributing not only to food production but also to economic growth. Early identification of diseases in plants presents a significant challenge, as timely detection is crucial to increasing agricultural production. Incorporating the latest technology, such as artificial intelligence (AI) techniques, in fields can reduce major losses for farmers and also improve productivity. In this paper, we have proposed a Deep Vision transformer and an Explainable AI-based technique to overcome the various diseases in plants. Here, we have used the dataset "Plant Village," as a case study that focuses on two majorly grown crops like: potatoes and tomatoes. Further, we have analyzed nine diseases that affect these crops around the globe, with tomatoes showing six different conditions and potatoes affected by three bacterial diseases. The proposed DVTXAI framework incorporates the Deep vision transforms to detect plant diseases at an early stage. In the experiments, the proposed model achieves an accuracy of 93. 56% for tomatoes and 99.95% for potatoes. Additionally, the model is Explainable AI (XAI), which reveals the transparency mechanism that helps the farmer to make the right decision at an early stage of the crop.

Keywords: AI in agriculture, Explainable AI, Plant disease detection, transformer learning, SHAP

---

1. Introduction

In light of ongoing population expansion and limited resources, it is urgently necessary to consistently supervise agricultural operations using AI/ML. This is vital to optimize food production efficiency while protecting natural ecosystems. Agriculture is one of the cornerstones of human civilization, providing food, livelihoods, and economic stability to

*Corresponding author

societies worldwide. From vast fields of staple crops to specialized plantations and agro-industries, agriculture encompasses a diverse array of practices, cultures, and ecosystems. sustainability. In many countries, agriculture forms the backbone of their economies, providing food to the people and raw materials to various industries. The challenges faced by the agricultural sector are multifaceted, ranging from climate change-induced changes in the growing seasons to the evolving demands of a growing global population. The ability to consistently produce high-quality crops is closely intertwined with a nation's capacity to ensure the health and vitality of its plants.

In India, 58% of the total population is mainly dependent on agriculture for their livelihoods, as stated by the Ministry of Agriculture and Farmers Welfare. The loss of crop production is often attributed to factors such as weeds, pests, and diseases, which are responsible for 15-25% of the total loss of crop production in India. The demand for efficient farming processes in agriculture and the food industries is rapidly increasing, but various plant diseases affect this process. Detecting and addressing plant diseases in large crop fields is a complex task that requires the use of optical observation of leaves and the expertise of skilled manpower [1, 2, 3]. This task is particularly challenging for both farmers and individuals without experience in the field, as visual judgment of disease classification can contain bias, optical misinterpretations, and errors. The use of image processing tools has become necessary to facilitate the detection of plant diseases [4]. Making use of image processing and computer vision techniques, we can identify plant diseases in their early stages.

Several researchers and scientists have identified the salient issues and challenges associated with the analysis of plant diseases [5, 6].

1.1. Plant disease and its symptoms

Plant diseases emerge as consequences of disturbances in the structure, physiology, or behavior of plants, which lead to troublesome conditions. The root causes of these diseases are multifaceted, arising from infectious agents such as fungi, bacteria, and viruses, as well as non-infectious factors such as sunburn and mineral deficiencies. Diseases originating from infectious agents are classified as biotic diseases, while those originating from noninfectious factors are termed abiotic diseases. The distinction between them lies in the degree of risk they pose, with biotic diseases, propelled by infectious agents, presenting a more formidable threat, whereas abiotic diseases are relatively less concerning due to their limited potential for transmission.

In contrast, the research attention bestowed on viral infections has been relatively modest. The spectrum of commonly encountered diseases, including blight, mildew, rust, spots (attributed to fungi or bacteria), and canker, has received significant attention within studies on the classification and identification of diseases [7, 8]. Expanding the horizon, investigations have ventured into realms that explore plant diseases rooted in nutrient deficiency, along with the intricate relationship between nutrient scarcity and its implications for disease resistance and tolerance.

As the pursuit of automated plant disease detection gains momentum, often leveraging sophisticated computer vision techniques, the foundation of success rests upon meticulous feature selection and extraction, coupled with the discerning choice of classifiers for precise disease classification. These efforts are based on the customization of the selection of features to align with the unique symptoms exhibited by different diseases in diverse plant categories.

1.2. *Significance of Deep Learning in Plant Disease Detection*

Traditional disease detection methods have proven to be labor intensive, time consuming, and often prone to inaccuracies. In this context, the emergence of deep learning techniques has revolutionized the field of plant disease detection, offering promising solutions to more efficient and accurate disease diagnosis.

Deep learning, a subset of machine learning, has garnered immense attention in diverse applications due to its ability to automatically learn intricate patterns and features from large-scale data [9, 10]. It has proven its mettle in image recognition, natural language processing, and speech recognition and has now found its way into agriculture to tackle challenges in plant disease detection. Unlike traditional machine learning techniques, deep learning models, especially Convolutional Neural Networks (CNNs), have the inherent ability to automatically extract hierarchical representations from raw data, which makes them well suited for image analysis tasks [11]. This property makes CNNs particularly effective in identifying complex patterns and subtle visual cues present in images of diseased plants, allowing accurate disease identification and detection [12]. The use of deep learning techniques for the detection of plant disease offers numerous advantages. Firstly, it significantly reduces the time and effort required for diagnosis, allowing for early detection and rapid response to disease outbreaks. Secondly, deep learning models have demonstrated remarkable accuracy rates, often exceeding those of human experts in disease identification. This enhanced precision minimizes misdiagnoses, which is crucial for minimizing unnecessary pesticide use and its associated environmental impacts.

Recent studies in image processing have shown that Vision Transformers (ViTs) and other hybrid architectures combining ViTs with CNNs outperform traditional CNNs in several key areas. Vision Transformers[13], leveraging self-attention mechanisms, excel in handling images with noise and augmentation by maintaining robust performance under varied conditions. They offer a comprehensive analysis of image data from all layers, unlike CNNs, which may lose spatial information through clustering operations. Furthermore, ViTs are more efficient, requiring fewer computational resources and less training time, particularly when fine-tuning. Although CNNs still demonstrate superior generalization with smaller datasets, ViTs gain an edge by effectively learning from fewer images due to their patch-based approach, which allows for a richer interplay of image features. This nuanced understanding of the strengths and weaknesses of each architecture paves the way for a more targeted and efficient use of these models in practical applications.

In the present work, we have proposed a vision image transformer learning model to explain the detection of plant disease together with SHAP, which will allow the identification of those who have leaf disease due to bacterial infection from environmental change. The proposed model eliminates the shortcomings of traditional neural network modes such as CNNs, capsule networks, etc. The main contributions of this paper can be summarised as follows:

1. We have applied the deep vision transformer model to detect diseases on potato and tomato plants.
2. We have obtained higher accuracy on multi classes, i.e., for potato three class values, and tomato, six class values.
3. We have compared and categorized the results into healthy and early blight diseases, early blight, and late blight.
4. To make the proposed model Explainable AI we have applied the SHAP library.
5. This research will assist new agriculture startups to better produce vegetables.

The remainder of the paper is organized into the following sections. Section 2 reviews with critical analysis different approaches to deep learning and machine learning to detect plant diseases. Section 3 explain the methodology comprising dataset description, augmentation, performance evaluation, etc. Section 4 presents the proposed framework of DeepViT for plant disease detection model and the explanation of each step of the framework. Section 5 presents the Algorithm and its steps. Section 6 provides the Explainable AI feature on DeepViT using SHAP algorithm to identify the dominant features. Section 7 presents the experimental results obtained and the related discussion. Finally, Section 8 concludes this paper.

2. Deep Learning in Plant Disease Detection: A brief review

Digital image processing techniques have been quite prominent in different AI applications, as they are capable of performing actions such as identifying the required focal point (set of pixels) of the image and its classification. Similarly, its applications are widely known in agriculture for the detection of plant diseases [14, 15, 16, 17, 18].

In recent times, image processing, machine learning, and deep learning for automatically detecting crop diseases have gained increasing attention and importance.

2.1. Disease detection via Machine Learning Classifiers

Researchers have explored various image processing and machine learning techniques to detect plant diseases.

In [19], The SVM-based image processing approach was used to analyze and classify prevalent diseases in rice crops, and achieved an accuracy of 94.16%. In the oil palm industry, notable economic losses caused by leaf diseases led to a study combining k-means clustering and a multiclass SVM classifier, achieving accuracies of 97%, and 95% respectively [20]. In [21], a method for the classification and detection of bean diseases was introduced, achieving an accuracy of 98. 6% combining image segmentation, feature extraction, and machine learning. For soybean culture, [22] proposed a semi-automatic leaf disease detection and classification system using image processing techniques, providing a practical solution for early detection and management of plant health issues. In [23], a neural network-based method was developed for the identification of plant diseases, achieving impressive prediction accuracy, reaching 100% for wheat and grape diseases. In [24], used the color, texture and shape features with the SVM classifier to achieve an accuracy of 96.3% for the detection of plant diseases. In[25], introduced a method to recognize cucumber diseases using superpixels, the expectation maximization (EM) algorithm, and the features of PHOG, achieving a precision of 94.29% in a leaf image database with cucumber diseases. Sannakki et al. [26] used neural networks to diagnose and classify grape leaf diseases with an accuracy of 92.11%. In [27], a fuzzy logic-based method was proposed to detect and classify leaf diseases with an accuracy of 91.25%. In [28] used image processing techniques and the SVM classifier, achieving an accuracy of 90.63% to detect and classify plant leaf diseases. In [29], applied K-means clustering, contour tracing and machine learning approaches (SVM, K-NN, and CNN) to classify tomato leaf diseases, demonstrating effective performance in real-time processing scenarios. CNNs were also used as feature descriptors in combination with SVMs in another study [30]. In[31], presented a survey of the techniques used to classify diseases in plant leaves, achieving an average accuracy of 97.6% with an image segmentation algorithm.

Another survey [18] provided an overview of key phases in the detection and classification of plant diseases, achieving an overall precision of around 93% using the k-means segmentation technique.

Similarly, non-black-box classifiers - Random Forest and Histogram of Oriented Gradients (HOG) - were used to detect papaya leaf diseases [32]. Despite the challenges posed by variations in leaf images, the approach achieved a precision of 70%.

However, all of the above approaches suffer from a common and inherent drawback of Non-Deep Learning approaches, i.e. manual feature extraction. Manual feature extraction is a very tedious task, increases the complexity of data preparation, and more importantly makes it difficult to extend the current study to cover more disease classes, as that might require us to do a completely new feature extraction process from scratch.

2.2. Disease detection via Deep Learning

In recent studies, several researchers have explored the potential of deep learning techniques for the identification and detection of plant diseases. Barbedo et al. [33] investigated using individual lesions and spots instead of whole leaves, resulting in 12% higher accuracies compared to the original images. This approach not only increased data variability, but also allowed the identification of multiple diseases on the same leaf. Similarly, Coulibaly et al. [34] proposed a deep learning approach with transfer learning for the identification of mildew diseases in pearl millet crops, achieving a promising precision of 95 00% and providing practical and efficient data analysis capabilities. Fuentes et al. [35] introduced a robust deep learning-based detector for the real-time recognition of tomato plant diseases and pests, utilizing various deep meta-architectures and achieving effective recognition even in complex scenarios. Furthermore, Kaya et al. [36] explored the impact of transfer learning on deep neural network-based plant classification models, demonstrating its significant potential in improving automated plant identification systems. Furthermore, Ferentinos et al. [37] developed convolutional neural network models for plant disease detection, achieving an impressive 99.53% success rate in identifying combinations [plant, disease]. This high success rate suggests the model's utility as an effective advisory tool and its potential for integration into real cultivation disease identification systems.

Roy et al. [38] introduced a meticulously crafted PCA DeepNet along with a custom CNN classifier, achieving an impressive classification accuracy of 99.60% and a mean precision of 98.55%. Similarly, deep learning techniques have been applied to diverse crop species, such as sunflower crops in Italy [39], where deep CNN and vegetation indices were used to improve segmentation performance, evaluated using the Mean Intersection Over Union (MIoU) metric. For crop disease identification, specific crops have been the focus of research, such as maize leaf diseases, where improved deep learning models based on GoogLeNet and Cifar10 achieved high average accuracies of 98.9% and 98.8%, respectively [40]. Other studies targeted individual crops, such as cardamom plant leaf diseases, using U2-Net for background removal and EfficientNetV2 for classification, resulting in a remarkable maximum detection accuracy of 98.26% [41]. In a similar vein, an effective deep CNN model achieved 98% classification accuracy to identify apple leaf diseases [42].

In the context of crop species discrimination, Eddy et al. [43] compared the neural networks (NN) and maximum likelihood (MLC) classifiers. NN classification showed slight improvements over MLC in all cases, indicating the potential of neural networks in this domain. Using publicly available data sets, Liu et al. [44] achieved an accuracy of 99.34% on a held-out test set using a deep convolutional neural network, demonstrating the feasibility of using deep learning for the classification of diseased and healthy plant leaves. Furthermore, the integration of neural networks (CNNs) has proven effective in identifying diseases in citrus fruits and leaves [45]. The proposed CNN model achieved a high test accuracy of 94.55%, making it a valuable decision support tool for farmers looking to classify citrus fruit and leaf diseases.

In addition, novel deep learning models have been developed to address plant disease identification. Hassan et al. [46] introduced a novel model based on the initial layer and the residual connection. Separable depth-wise convolution was utilized to reduce the number of parameters, and the model achieved high precision on three different plant disease datasets. Zhou et al. [47] proposed a restructured residual dense network (RDN) for the identification of tomato leaf disease, combining the strengths of deep residual networks (ResNets) and dense networks. The RDN achieved an impressive top-1 average identification accuracy of 95% in the Tomato test data set in the AI Challenger 2018 data sets.

3. Methodology

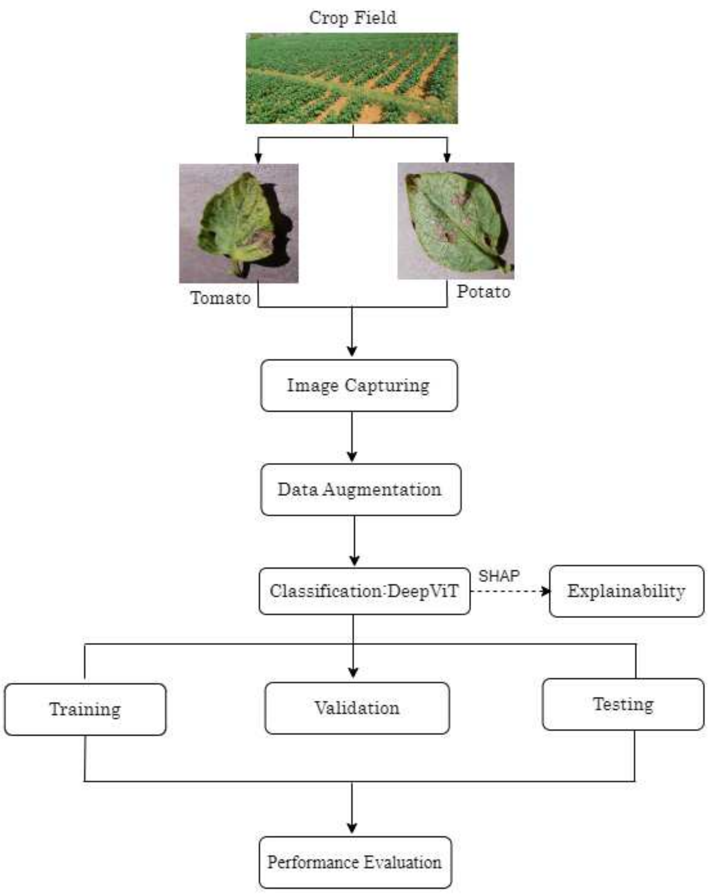


Figure 1: Distribution of Images among classes in Potato Dataset

In this section, we have presented the proposed model for detecting plant diseases. This framework utilizes image data sets for both potato and tomato, as depicted in Figure 1. The process consists of four steps that include the capture of images, data enrichment, implementation of the proposed DeepViT framework, and an AI explainable model using SHAP.

3.1. Dataset Description

The PlantVillage dataset [48] consists of images of many different crops. However, for experimentation purposes, we have selected images of tomato and potato crops. These selections are based on their agricultural importance and the prevalence of various diseases within these species, making them ideal crops to assess the efficacy of advanced machine learning approaches in disease detection and classification. This dataset is also known for its extensive collection of images of plant leaves documenting various states of disease. Initially, it had 54,305 images; after data augmentation processes such as image flipping, gamma correction, noise injection, PCA color augmentation, rotation, and scaling, it has a total of 61,486 images [49]. From the data set taken, the tomato data set includes images of 5 different diseases, together with images of healthy leaves, as shown in Figure 3. Similarly, for the potato dataset, there are three classes, two distinct diseases, and a healthy class, as demonstrated in Figure 4. The class distribution for tomatoes and potatoes diseases within the data set is shown in Table 1. Table 2 defines the class values distribution in potato dataset, and Table 3 defines the class value distribution in potato dataset.

Table 1: Dataset description: Total number of images in each class

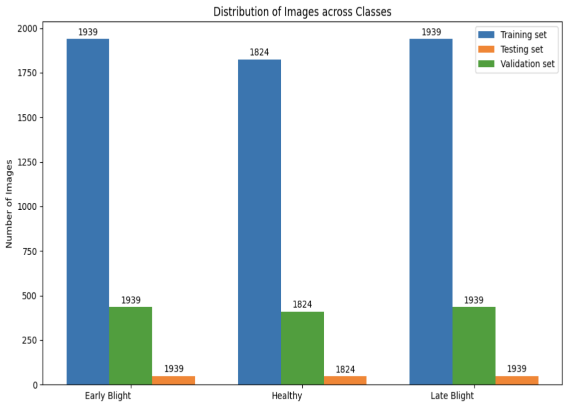
Classes	Number of Images
Tomato Healthy	2407
Tomato Early Blight	2400
Tomato Late Blight	2314
Tomato Bacterial Spot	2130
Tomato Septoria Leaf Spot	2181
Tomato Yellow Leaf Curl Virus	2451
Potato Healthy	2283
Potato Early Blight	2424
Potato Late Blight	2421

Table 2: Distribution of Images in Potato Dataset

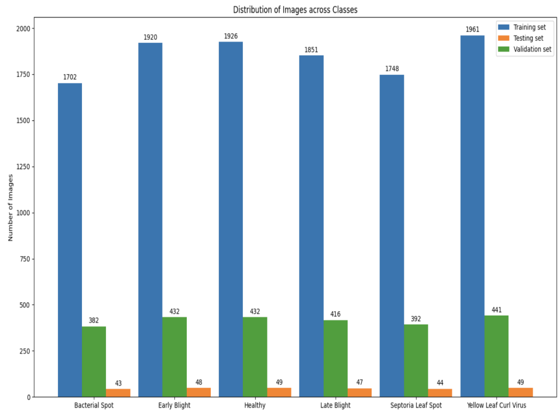
Classes	Train	Valid	Test
Potato Early Blight	1939	436	49
Potato Late Blight	1939	436	46
Potato Healthy	1824	410	49

Table 3: Distribution of Images in Tomato Dataset

Classes	**Train**	**Valid**	**Test**
Tomato Healthy	1939	436	49
Tomato Early Blight	1939	436	48
Tomato Late Blight	1824	410	47
Tomato Bacterial Spot	1824	410	43
Tomato Yellow Leaf Curl Virus	1824	410	49
Tomato Septoria Leaf Spot	1824	410	44



(a)



(b)

Figure 2: Distribution of Images among classes in (a) potato, (b) tomato

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: Samples of Tomato Leaf Diseases (a) Healthy, (b) Early Blight, (c) Late Blight, (d) Bacterial Spot, (e) Septoria Leaf Spot, (f) Yellow leaf Curl Virus

3.2. Data pre-processing

Upon completion of the data collection, a comprehensive preprocessing routine has been implemented to ensure the uniformity and quality of the data set. This helps reduce potential sources of bias and variance, thus improving the reliability of sub-analyses. For this, we have created a Python script using 'pytorch' library package to image samples. This script enables the scaling of all the collected images to a specific dimension of 256 x 256 pixels. This resizing was critical to normalizing the dimensions of the input data,

(a) (b) (c)

Figure 4: Samples of Potato Leaf Diseases (a) Healthy, (b) Early Blight, (c) Late Blight

facilitating consistent processing and analysis throughout the entire data set.

3.3. Data Augmentation

The input dataset size is crucial for evaluating the performance of deep learning models. Insufficient input data can lead to overfitting, resulting in poor validation results. Therefore, we need to increase the dataset synthetically by applying different data augmentation methods, such as flipping, rotating, etc. This diversification of the data set enhances the robustness, generalization, and applicability of the models and algorithms trained on it.

3.4. Transformer Architectures: ViT and DeepViT

Transformers process input sequences—typically token embeddings—through multiple layers, each consisting of a multi-head self-attention mechanism that uses dot-product attention to integrate information across tokens, and a token-wise feed-forward layer (MLP). Both components are enhanced by layer normalization and residual connections to improve learning dynamics and stability.

The Vision Transformer (ViT) [51] employs the same Transformer architecture previously mentioned, but introduces a distinct approach in its image preprocessing layer. In order to capture global dependencies in the image, ViT substitutes self-attention techniques for the conventional convolutional layers. This method outperforms traditional CNN-based models in certain tasks, as evidenced by its encouraging performance in some benchmark datasets [52, 50]. The VIT architecture is made up of various significant layers, and an image is divided into a series of non-overlapping patches, which are then linearly projected based on a learned transformation. This makes it possible for the model to process the image more effectively. Furthermore, ViT incorporates a transformer encoder that makes use of self-attention methods in order to effectively represent features by capturing long-range dependencies between patches. ViT has a special CLS token at the beginning of the input sequence, which is used to derive the representation needed for the final classification task.

In the rapidly evolving field of deep learning, the introduction of the Deep Vision Transformer (DeepViT) marks a significant development in the field of image classification. Building upon the foundational Vision Transformer (ViT), DeepViT integrates a cutting-edge mechanism known as Re-attention, as shown in Figure 5. It effectively replaces the conventional layers of self-attention. This modification plays a significant

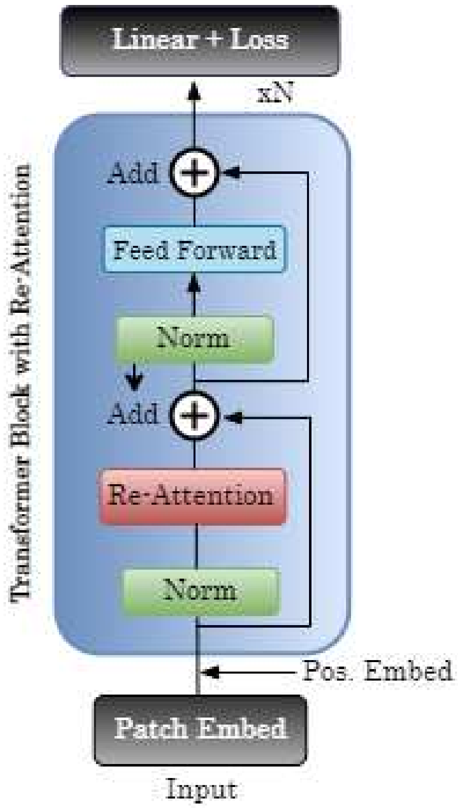


Figure 5: Basic Architecture of DeepViT [50]

role in addressing the attention-collapse problem that commonly occurs when scaling ViT architectures to increased depths [50]. The Re-attention mechanism regenerates the attention maps in the multi-head self-attention (MHSA) framework, promoting dynamic and learnable interactions among the various attention heads. This enhances the architecture's ability to discern complex relationships within images.

3.5. Performance Evaluation

Confusion matrix has been widely used to evaluate classification problems as it provides measures such as accuracy, precision, recall, f-measure, etc. They play a significant role in the evaluation of the model based on the data set and the problem specification. It is a 2×2 matrix Each cell of the matrix represents the model detection rate which is represented by – True Positive (TP), True Negative (TN), False Positive(FP), and False Negative(FN) [53, 54].

Using these four performance evaluation parameters, Equations 1, 2, 3 and 4 are calculated, each representing the accuracy, sensitivity, and specificity of the model, respectively.

1. **True Positive (TP):** If a plant is infected with the disease, the Deep-ViT classifier also detects it as infected.
2. **True Negative (TN):** if a plant is not infected with a disease and the Deep-ViT classifier is also detectable as not infected.
3. **False Positive (FP):** If a plant is not infected with the disease, the Deep-ViT classifier is also detected as infected.
4. **False Negative (FN):** If a plant is infected with disease, the Deep-ViT classifier also detects it as not infected.

Accuracy: It quantifies the proportion of correctly classified pixels relative to the total pixel count in an image. Although this metric provides a global measure of model accuracy, it does not account for class imbalances, which could skew the interpretation of model performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{1}$$

Precision: ratio of true positive observations to total predicted positives. This metric is crucial in determining the accuracy of the model in identifying only relevant instances.

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{2}$$

where TP denotes True Positives and FP denotes False Positives.

Recall: Recall, also known as Sensitivity, measures the proportion of actual positives that are correctly identified by the model. This metric is essential for assessing the completeness of positive predictions.

$$Recall/Senstivity = \frac{TP}{TP + FN} \times 100\% \tag{3}$$

where FN represents False Negatives.

F-measure: The F1-Score combines precision and recall into a single metric, offering a balanced assessment of segmentation performance.

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \times 100\% \tag{4}$$

3.5.1. Receiver Operating Characteristic (ROC)

In this study, the performance of plant village dataset has been evaluated by calculating the area under the *ROC* curve, which is known as the area under the curve (AUC).

The effectiveness of the suggested model was assessed using various metrics, including a confusion matrix, sensitivity, specificity, and precision. Basically, it is the trade-off between sensitivity and specificity [55]. The calculated AUC value of the curve is between 0 and 1. An AUC value close to 1.0 indicates that the model has high sensitivity and specificity [56]. Due to its robust assessment capabilities, the ROC is widely used in various fields [57, 58, 59]. The AUC of the ROC is estimated using the trapezoidal formula as given in Equation 5.

$$T(p,q,n) = \left(\frac{q-p}{n}\right) \times \left(\frac{f(p)-f(q)}{2}\right) \times \left(\frac{f(p)-f(q-p)}{n}\right) \tag{5}$$

The trapezoidal rule is a method used to approximate the integral of a function, $\int_p^q f(x)dx$, by dividing the interval from p to q into equal segments n. Each segment is used to form a trapezoid, the area of which is calculated and summed to provide an estimation of the area under the curve (AUC). This method defines each segment endpoint as $x_0, x_1, \ldots, x_n$ with $x_0 = p$, $x_n = q$, and each intermediate point $x_r = x_0 + r(q-p)/n$. The function $T(p,q,n)$, described in equation 5 and cited in [60], utilizes these calculations. The AUC, thus calculated, serves as a robust measure for evaluating the accuracy of Receiver Operating Characteristics (ROC) derived from a model.

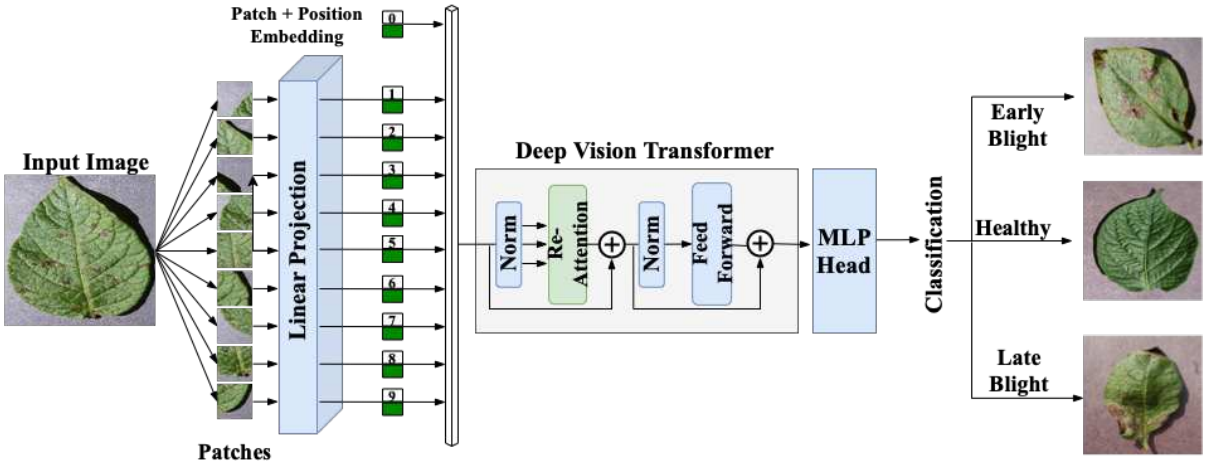


Figure 6: Proposed Framework for Potato and tomato Disease classification

4. Proposed Framework: DVTXAI

In the proposed Deep-ViT model, we have performed a parametric analysis evaluating the changes in patch sizes, the number of attention heads, and the depth of the transformer blocks affecting the model. Following is the flow of proposed model:

(i) **Input image patches:** The proposed model initially segments the input dataset images—specifically, potatoes and tomatoes—into smaller sections. Each image, with dimensions of 256 $\times$ 256 pixels, is divided into non-overlapping patches of 32 $\times$ 32 pixels. This segmentation strategy serves to simplify the processing of large image inputs while preserving the capacity to model intricate spatial interactions. Patch size can be adjusted according to the hyperparameter settings that consider the dimensions of the input image.: rewrite this for research paper.

(ii) **Patch embedding:** After patches, they are converted through the linear embedding process into a sequence of vectors suitable for processing by the transformer. This setup involves a patch embedding layer that acts as a translator, converting the image patches into a numerical sequence so that the transformer can interpret them. This layer includes a linear transformation, a ReLU function is used to enhance the practical utility of the numbers, and a dropout layer to prevent overfitting by omitting random information during training, this helps the model to manage unpredictability and improve its predictive accuracy.

(iii) **Positional encoding:** It is a technique that enhances each input vector within a sequence by integrating a fixed vector defined by its position. The integration of positional encoding within the Vision Transformer (ViT) model serves the purpose of retaining the spatial information present in the input image and improves the contextual meaning of the patch embeddings.

(iv) **Deep vision Transformer:** The Deep ViT block plays a crucial role in the proposed architecture by managing the sequence of patch embeddings and effectively capturing their interconnections. The encoder block consists of three components: a re-attention layer, a feed-forward neural network (FFNN), and additional multi-head re-attention layers. The multi-head attention layers are integral to the transformer architecture, helping the model understand the relationships between different parts of the input sequence of patches.

In Equation 6 defines the basic operation of the scaled dot-product attention. where the attention function calculates the dot products of the query Q with keys K, normalizes these by the square root of the key dimensionality d_k, and applies softmax to get the weights for the values V. This normalization stabilizes training gradients and allows the model to focus dynamically on different parts of the input, which is essential for understanding the input image. Equation 7 defines multi-head attention allows each "head" to process the input with distinct sets of learned projections W_i^Q, W_i^K, and W_i^V. This configuration enables the model to capture diverse features from different subspaces, enhancing its ability to represent and focus on various aspects of the input sequence comprehensively. Fianlly different multiple attention heads are then combined by concatenating their outputs, which is then multiplied by an additional learned projection W^O as shown in Equation 8. The multi-head self-attention mechanism is a key component of transformer architectures, enabling them to effectively handle a wide range of complex tasks, such as machine translation, text summarization, and context-sensitive information retrieval.

In this finding, a multi-head attention mechanism with four heads works better on our dataset.This allows the model to process various aspects of the patches simultaneously, enhancing the representation of features. The re-attention layer measures the similarity between elements in the sequence, creating a weighted combination of this information. This process is repeated multiple times, starting from different points to explore similarities. The results are then concatenated and linearly transformed to produce the final outcome. This is then passed on to the next step in the feedforward layer, where each neuron in one layer is connected to every neuron in the same layer. This consists of two linear layers connected by a Rectified Linear Unit (ReLU) activation function. This design helps the network to effectively combine and represent the non-linear relationships between the input patches.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{6}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{7}$$

$$\text{MSA}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{8}$$

4.1. Classification layer

The Vision Transformer (ViT) model generates its results by taking the output from the second multihead reattention layer, passing it through a linear layer, and then applying a softmax activation function. The softmax function converts the input into a probability distribution between multiple classes, allowing the model to provide a likelihood distribution that encompasses potential outcomes.

In the present work, we have performed the classification with multi-headed self-attention mechanism on the dataset that comprises four heads. Using multiple attention heads in transformers enhances their expressive power and modeling capabilities. This allows the model to process various aspects of the patches simultaneously, enhancing the representation of features. Additionally, a configuration with four sequentially arranged transformer blocks emerged as the most effective, striking a balance between model complexity and performance. This customized setup of the DeepViT model, which consists of four attention heads and four transformer blocks, showed superior performance, confirming its effectiveness for the characteristics of our data set and the designated classification tasks.

Similar to how a typical transformer breaks down a string of text into words, a vision transformer breaks down the input image into patches. Patches are equal-sized squares of pixels; thus, during patching the image is broken down into multiple patches. This is done to reduce the computational complexity while retaining the quality of the image with any given input size. In order to input these patches into the encoder, the patches are flattened into a single low-dimensional vector. While this flattening allows the Vision Transformer to treat patches as a Transformer would treat words (as a low-dimensional vector), it also results in a loss of positional information, i.e., the spatial order within the patches. This issue is addressed by applying positional encoding to the patches; this is done via sinusoidal encoding, wherein sine and cosine functions are used to create a position-specific vector for each patch; this vector stores the relative location of the patch within the image.

Further analysis was performed to compare the classification results between specific pairs of subclasses within each crop category. Notably, the results for the pairs of 'healthy' versus 'early blight' and 'early blight' versus 'late blight' were analyzed to assess the discriminative ability of the model between closely related disease states and healthy specimens. This comparative approach provided deeper insight into the efficacy of the model and potential areas for improvement in distinguishing between subtle variations in plant health and disease manifestations.

5. Algorithm: Deep Vision Transformer

Algorithm 1: Deep Vision Transformer

Input:
  `input_image`: RGB image to be processed
  `patch_size`: Size of the square patches (e.g., 16x16 or 32x32)
  `embedding_dim`: Dimension of the linear projection
  `num_heads`: Number of heads in Multi-Head Self-Attention
  `num_layers`: Number of transformer layers
  `ff_hidden_dim`: Dimension of hidden layers in Feedforward Network
  `num_classes`: Number of output classes (for classification)
  `position_encoding`: Positional encoding to add to patch embeddings
Output:
  `class_scores`: Prediction scores for each class
Steps:
1. Patch Embedding:
  Divide `input_image` into non-overlapping patches of size `patch_size`.
  Flatten each patch and project it to `embedding_dim` using a linear layer.
  Store the projected patches in `patch_embeddings`.
2. Positional Encoding:
  Create positional encodings based on the number of patches.
  Add positional encodings to `patch_embeddings`.
3. Class Token:
  Initialize a special `class_token` with `embedding_dim`.
  Insert `class_token` at the beginning of `patch_embeddings`.
4. Transformer Encoder: for *each transformer layer from 1 to* `num_layers` **do**
      end
      Perform Multi-Head Self-Attention:
  Apply multi-head self-attention on `patch_embeddings`.
  Apply Layer Normalization and Residual Connections.
Perform Feedforward Neural Network:
  Apply linear transformations and activations.
  Apply Layer Normalization and Residual Connections.
5. Output Processing:
Extract `class_token` from `patch_embeddings`.
Pass `class_token` through a linear layer to get the final `class_scores`.

The *Deep Vision Transformer* (ViT) algorithm is an advanced model designed for image classification tasks using a transformer-based architecture. It begins by taking an *input* The RGB image is divided into non-overlapping square patches of a specified size, such as 16×16 or 32×32 pixels. Each patch is then flattened to a vector and projected into a lower-dimensional space, defined by the *embedding dimension*, using a linear layer. These projected vectors, known as *patch embeddings*, are stored sequentially.

To retain spatial information about the position of each patch within the original image, *positional encodings* are generated and added to the *patch embeddings*. A unique learnable *class token*, which has the same dimensionality as the embeddings, is initialized and prepended to the sequence of *patch embeddings*. This *class token* plays a crucial role in the classification process, as it eventually holds the aggregate information needed for prediction.

The heart of the ViT algorithm lies in the transformer encoder, which consists of multiple layers specified by the user. Each layer within the transformer encoder performs several key operations. First, *multihead self-attention* is applied to the *patch embeddings*, allowing the model to simultaneously focus on different parts of the image and capture complex relationships between patches. Following self-attention, *layer normalization* and *residual connections* are employed to stabilize and enhance the learning process. Next, a *feedforward neural network*, which includes linear transformations and activation functions, is applied to the output of the self-attention mechanism. This is followed again by *layer normalization* and *residual connections*, ensuring that the transformations are effectively integrated.

After passing through all transformer layers, the *class token* is extracted from the sequence of *patch embeddings*. This token, now enriched with information from the entire image, is fed through a final linear layer to produce the *class scores*. These scores represent the model predictions, indicating the likelihood that the input image belongs to each possible class.

6. Explainable DeepViT using SHAP

6.1. General Introduction to SHAP

SHAP (SHapley Additive exPlanations) uses principles from cooperative game theory to explain the predictions of machine learning models. By drawing from the concept of Shapley values, which allocate ”payouts” to players based on their contribution to the collective success in game theory, SHAP assigns an importance value to each feature based on its contribution to a prediction. This methodology, first introduced by Lundberg et al. in 2017 [61], offers a consistent and locally accurate explanation framework, providing insight into the general importance of characteristics across the model as well as detailed explanations for individual predictions.

The methodology operates by calculating the average marginal contribution of each characteristic to the predicted outcome. This is particularly illustrative in tasks like image classification, where the impact of each pixel is visualized through scores: red pixels typically indicate an increase in the probability of correct classification, whereas blue pixels indicate a decrease. Such detailed pixel-based analysis is crucial in applications such as diagnosing diseases in mulberry leaves, where understanding of feature contributions guides better model interpretations.

The formal computation of SHAP values is rooted in the Shapley value formula from cooperative game theory:

$$\phi_k = \sum_{M \subseteq N \setminus k} \frac{|M|!(|N| - |M| - 1)!}{|N|!} \left[f_x(M \cup k) - f_x(M) \right] \tag{9}$$

Here, f_x denotes the change in model output attributable to the inclusion of feature subsets. The subset M consists of all features except the feature k, and the factorial terms weigh the permutations of the features within M.

The expected prediction $f_x(M)$, pivotal in deriving ϕ_k, is defined as follows:

$$f_x(M) = \mathbb{P} \left[f(x) \mid x_M \right] \tag{10}$$

To simplify the implementation of SHAP, each feature x_k in the model is replaced with a binary indicator b_k^0, which marks the presence or absence of x_k. This leads to a simplified model representation:

$$l(b^0) = \phi_0 + \sum_{k=1}^{A} \phi_k b_k^0 \tag{11}$$

In this equation, ϕ_0 represents the bias term, $\phi_k b_k^0$ quantifies the individual feature contributions, and A denotes the total number of features simplified in the model. This substitute model, $l(b^0)$, functions as a proxy for the original model $f(x)$, facilitating a deeper understanding of how each feature influences the model predictions.

6.2. Application of SHAP on Plant Disease Detection

Our research strives to advance the interpretability of AI within the sphere of plant disease identification, using Transformer learning models. The intricacies of Transformer architectures necessitate a comprehensive understanding of their decision-making processes, especially for applications aimed at discerning specific indications of disease within plant imagery.

We have developed a Vision Transformer (ViT) model, trained on a data set composed of plant images, to categorize these images into distinct classes indicative of various plant diseases. The model employs an efficient transformer mechanism that is adept at processing the sequential nature of image patches, thereby capturing the nuanced local and global dependencies present within the image data.

To enhance the explainability of our ViT model, we have employed SHAP (SHapley Additive exPlanations), aiming to elucidate the elements within input images that significantly influence the model's diagnostic predictions. This approach involves calculating SHAP values, which measure the impact of each image patch (or pixel, for more detailed explanations) on the model's categorization output.

Adapting SHAP for our model involved tailoring the explanation generation process to accommodate the Vision Transformer's architecture. This necessitated modifications to target the output of intermediate or final layers for a comprehensive explanation. Using GradientExplainer, a SHAP variant optimized for models with inputs directly influencing their differentiable output, we generated SHAP values for a select portion of our test dataset, which included representations of various plant diseases.

The visualization of these SHAP values provided profound insight into the model's decision-making criteria. Preliminary analysis suggested that specific image features, such as textures, colors, and patterns, play pivotal roles in the model's ability to detect certain plant diseases. These findings not only corroborate the model's validity with agronomic principles, but also unveil novel patterns the model has discerned.

The elucidations provided by SHAP have significant practical implications for deploying our AI model in real-world scenarios. They facilitate expert verification of the model's judgments, substantiate the model's dependability, and offer guidance for model optimization by identifying potential biases or improvement avenues.

6.3. Explainability through SHAP

When we applied SHAP to our DeepViT model using their Gradient Explainer, we observed two notable properties of the SHAP value map we obtained. First, aside from the areas of the leaves that have disease spots on their surface the SHAP values also created clusters near the boundary of the leaves and at the corner of the images. Second, the shape values when plotted on a graph appeared as squares with visible and distinctive boundaries in some cases. Upon further observation, we discovered that the dimensions of the squares correspond to the patch size of our vision transformer model. The above observations prompted us to adjust the way we had implemented SHAP for our DeepViT model, wherein we got some remarkable results when we took the mean of the SHAP values over the patches of the image. As per the colour map that we have taken, blue represents negative SHAP values and red represents positive, additionally the intensity of the red or blue colour dictates the magnitude of the SHAP value. Thus a deep-red colour represents a highly positive SHAP value and a deep-blue colour represents a highly negative SHAP value. As we have taken the mean of the SHAP values over each patch of the image, our results show whether a given patch has a positive or negative contribution towards the classification of the image instance into a certain class, and depth of blue or red colour shows the magnitude of that contribution. Through this, we are able to outline which features of the image contribute towards classification into certain classes of diseases.

6.3.1. SHAP implementation on Potato Data

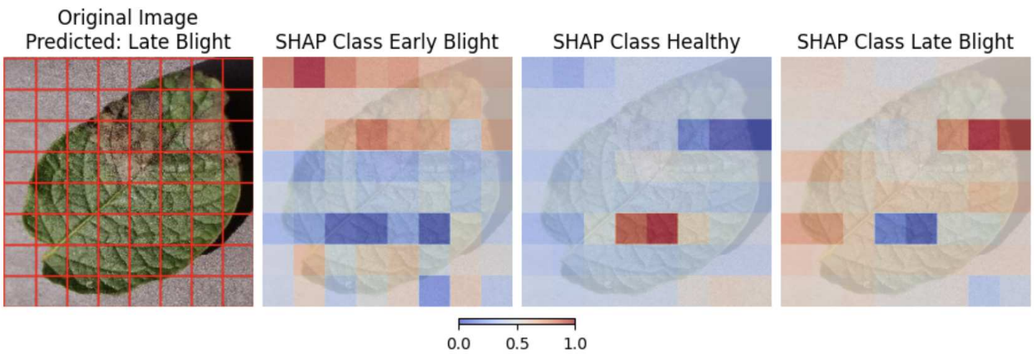


Figure 7: Example of SHAP output on Potato Data

In the given example, a random image is taken from the Potato Data of Plant Village Dataset, and its SHAP results as plotted and presented in Fig. 7. In the above, we can see that the region of the leaf with disease marks has negative SHAP values in the case of the Healthy class and positive SHAP values in the case of the two disease classes. Meanwhile, a patch of the leaf that looks completely fine and has no symptoms of a disease contributes with negative SHAP values in the case of the two disease classes. As for the diseased area, since it matches closer to the late-blight class as compared to the early blight class, the intensity of the red is deeper i.e. the magnitude of the positive SHAP values is higher for the correct class. In addition to this, the instability as mentioned before can also be seen as the patches in the background, on the corners, and along the borders also have some SHAP values associated with them. The colour bar just below the plots signifies the colour associated with positive and negative SHAP values, as we have normalised the SHAP values to a range between 0 and 1, the negative values now start from zeroes, the SHAP values with very low magnitude are normalised to 0.5 and 1 signifies positive SHAP values with high magnitude. Fig. 8 shows that the SHAP results in Potato Data are well explainable across all the classes.

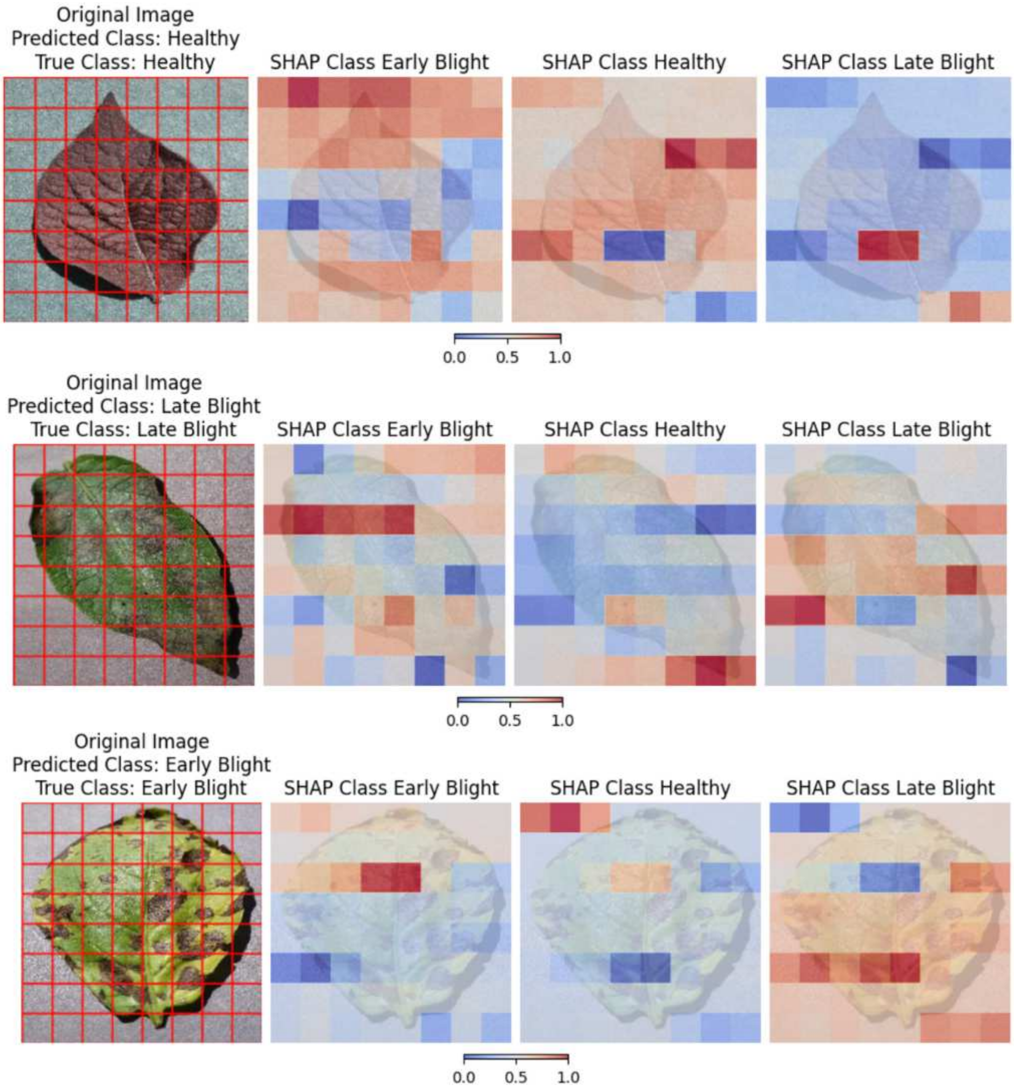


Figure 8: SHAP results across various classes of Potato Data

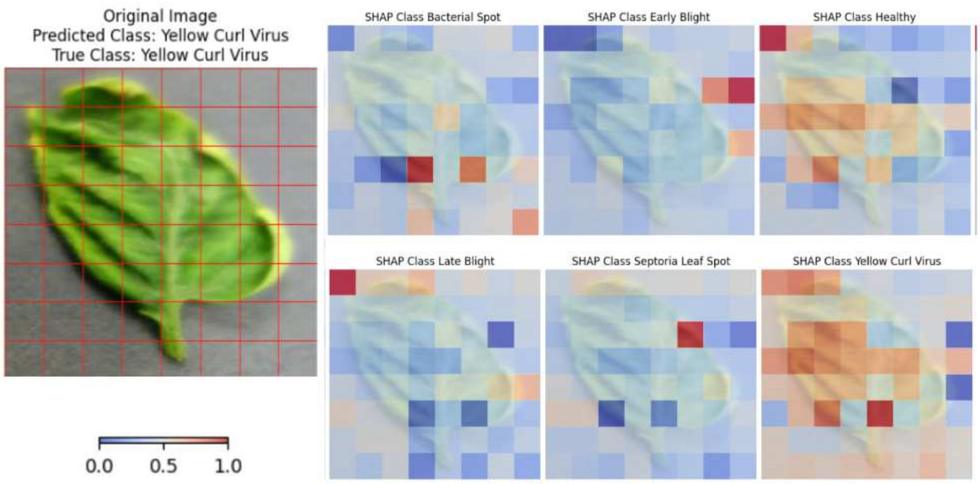


Figure 9: Example of SHAP output on Tomato Data

6.3.2. SHAP implementation on Tomato Data

Here we have plotted the SHAP graphs randomly for the images in the Tomato Data within the Plant Village Dataset. In this case, we have an example of the Tomato Yellow Curl Virus, when afflicted by this disease some notable changes occur to the plant leaves: they curl upwards towards the inside of the leaf, they become thicker, and have a leathery texture, finally, the leaves become yellow near the veins this change is a result of interveinal chlorosis. In the results plotted in Fig. 9, all disease classes except Yellow Curl Virus have negative SHAP values, signified by their blue colour, over the green part of the leaf, this is because unlike other diseases Yellow Curl Virus doesn't lead to the leaf developing any kinds of spots or dead area on its surface, thus the absence of these features result in negative SHAP values over their respective pixel areas. This is also the reason why these green patches contribute positively towards the 'Healthy' class classification, as these areas of the leaf's surface are blemish-free. What distinguishes between the 'Healthy' class and the 'Yellow Curl Virus' class is the upward curls at the peripheries of the leaf and a pattern of yellowing towards its centre - attributed to interveinal chlorosis. This is reflected in the respective SHAP plots where the red blocks - signifying positive values are limited to the greener patches on the leaf for the 'Healthy' class whereas the patches of the images that visually capture the yellow patterns about the leaf veins are marked by positive SHAP values in case of 'Yellow Curl Virus', which also has the patches capturing the upward-curled borders of the leaves distinguished by the red blocks. The results for the Tomato Data also suffer from the instability that we talked about earlier for the Potato Data, nevertheless, the SHAP results are interpretable, as seen above, and provide insight into how the different areas (patches to be precise) capture various features of the leaf's surface and contribute to its classification into the respective classes.

7. Experimental Results and Discussion

Table 4: Overall comparison between Tomato and Potato diseases

Plants	Accuracy	Precision	Recall	F1-Score
Tomato	93.56	93.58	93.54	93.55
Potato	97.55	97.53	97.54	97.53

Table 5: Performance comparison between Healthy and Early Blight

Plants	Accuracy	Precision	Recall	F1-Score
Tomato	93.56	93.58	93.54	93.55
Potato	99.15	99.15	99.15	99.15

Table 6: Performance comparison between Early Blight and Late Blight

Plants	Accuracy	Precision	Recall	F1-Score
Tomato	99.41	99.40	99.40	99.40
Potato	99.95	99.95	99.95	99.95

Table 7: Comparison of DeepVIT with other leading models for Tomato Data.

Model	Accuracy	Precision	Recall	F1-Score	Parameters (in Millions)	Data Augmentation
MaxViT [62]	97%	97%	97%	97%	300	Yes
LeNet [63]	94.0%	94.81%	94.78%	94.8%	-	Yes
ViT-1 [64]	98.51%	99.2%	99.3%	98.2%	-	Yes
ViT-SmartAgri [65]	90.99%	90%	89%	89%	13	Yes
DeepViT	93.56%	93.58 %	93.54%	93.55%	-	Yes

Table 8: Comparison of DeepVIT with other leading models for Potato Data.

Model	Accuracy	Precision	Recall	F1-Score	Parameters (in Millions)	Data Augmentation
EfficientRMT-Net [66]	97.6%	94.12%	94%	96.2%	8	Yes
SBCNN [67]	96.98%	94%	96%	95%	16	Yes
MobOca_Net [68]	97.73%	-	94.33%	94.33%	3.5	Yes
DeepVIT	99.95%	99.95%	99.95%	99.95%	3.5	Yes

The objective of the experiment is to detect different plant diseases using a deep learning model. For this, we proposed a deep vision transformer model and applied it to the publicly available Plant Village dataset [48]. The classification of diseases was performed on two different plants, namely, potato and tomoto. For the tomato dataset, classifications were performed in six subclasses: Bacterial spot, Early blight, Healthy, Late blight, Septoria leaf spot, and yellow leaf curl virus. However, the potato data set included three subclasses: Early blight, Healthy, and Late blight. These results were meticulously compared between different classes within the tomato and potato data sets, each comprising different subclasses and diseases.

For our experimental setup, we adopt a structured approach to data management, segmenting the 'Tomato' and 'Potato' images into training, validation, and testing processes. This segmentation follows a strategic split designed to optimize the learning process and evaluate the model's performance comprehensively. Through this focused application of the DeepVit model on Tomato and Potato diseases within the PlantVillage dataset, our research endeavors to advance the field of plant disease detection, offering insights into the transformative potential of transformer learning technologies in agricultural contexts.

The proposed model was developed using Deep Vision Transformer (Deep ViT) algorithms scripted in PyTorch. The supporting libraries used include Pandas, NumPy, Torchvision, Matplotlib, shap, etc. The model architecture begins by dividing each input image into fixed, nonoverlapping patches of size 32x32. These patches are then passed through a linear projection to create embeddings. Positional embeddings are added to preserve the spatial context of the patches, ensuring that the model retains critical information about the relative position of each patch in the image. These positional embedded patches are fed into the Deep ViT architecture, where Re-Attention replaces the traditional Multi-Head Attention from the original Vision Transformer. Re-Attention regenerates attention maps, enhancing their diversity across different transformer layers while adding minimal computational and memory overhead. In our implementation, we used 8 attention heads for Re-Attention. The output from the transformer layers is passed through a feedforward neural network with a depth of four layers. Finally, the output of Deep ViT is directed to a multilayer perceptron (MLP) with a total of 2,048 perceptrons for the final prediction. The model was trained using a learning rate of 3e-5 and a batch size of 64 for 20 epochs, providing a robust framework to process data sets and achieve effective results.

The overall performance of both the crops were presented in Table 4 and 7. Other two tables 5 and 6 show the performance of healthy and early blight, and early blight and late blight. The accuracy of the classification models was evaluated for each subclass within the respective datasets. The model achieved an accuracy of approximately 92 in classifying the six different tomato subclasses. In contrast, the classification of the three potato subclasses yielded a higher accuracy, approximately 98, underscoring a more robust model performance in this context.

The training and validation of both dataset were drafted in Figures 10 and 12 The AUC of ROC of the proposed model is 0.95, depicted in Fig. 11, which is much higher than the threshold level (i.e., 0.5). This is a clear indication of the least trade-off between sensitivity and specificity of the model and can be considered as a good grade of classification [69].

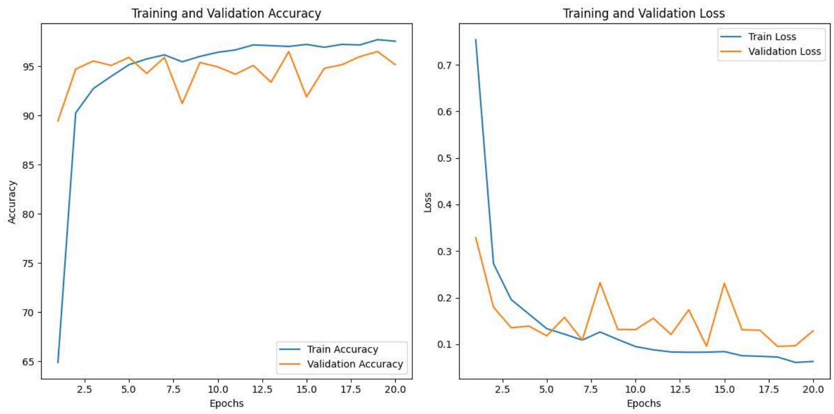


Figure 10: Potato Training and Validation accuracy

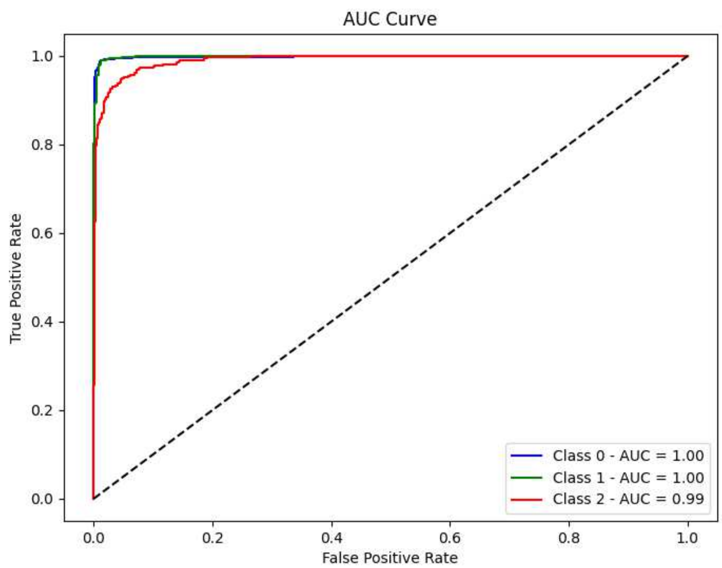


Figure 11: Potato ROC and AUC for the proposed Transformer Learning model

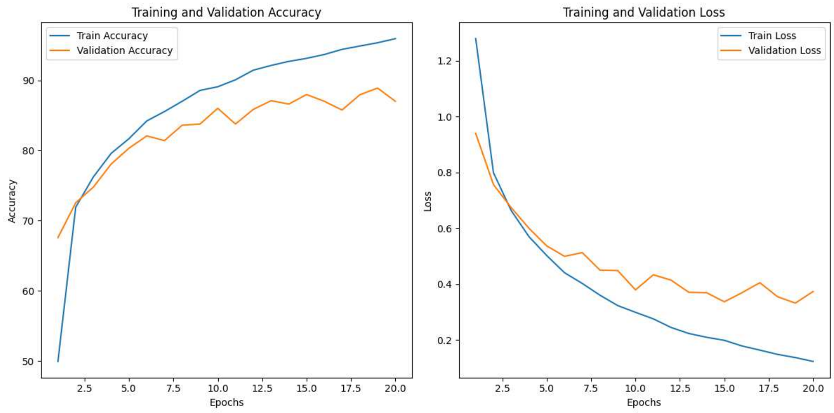
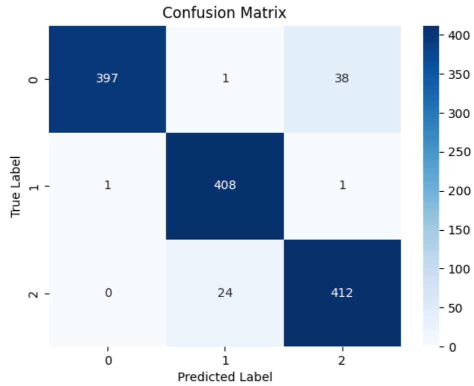
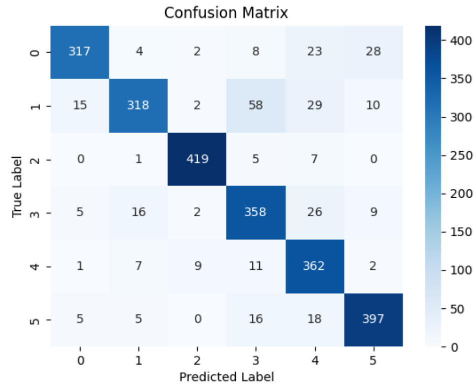


Figure 12: Tomato Training and Validation accuracy



(a) Confusion matrix for Potato



(b) Confusion matrix for Tomato

Figure 13: Confusion matrices for (a) potato and (b) tomato

8. Conclusion

This research work introduces deep vision transformer and explainable AI in the field of agriculture. The proposed architecture improves the early detection of plant diseases for individuals without expertise in plant pathology, by using the proposed DeepViT model to detect diseases in tomato and potato leaves. In the experiments, we have achieved accuracy rates of 93.56% for tomatoes and 99.95% for potatoes as shown in Tables 8 and 3. Additionally, we have incorporated the SHAP algorithm that makes the proposed model explainable and enhances the transparency of the model decision-making process. Future research will focus on implementing the model in a manner that reduces the need for manual involvement in identifying plant diseases. We will prioritize the development of a user-friendly system to streamline operations, particularly for the agricultural community. The model will be further refined and integrated with big data analytic tools in cloud computing services to monitor and analyze disease patterns in other crops, ensuring a comprehensive and efficient approach to plant disease management.

References

[1] U. Mokhtar, N. El-Bendary, A. E. Hassenian, E. Emary, M. A. Mahmoud, H. A. Hefny, M. F. Tolba, Svm-based detection of tomato leaves diseases, in: IEEE

Conf. on Intelligent Systems, 2014. URL: https://api.semanticscholar.org/CorpusID:2208239.

[2] S. Verma, A. Chug, A. P. Singh, Prediction models for identification and diagnosis of tomato plant diseases, in: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1557–1563. doi:10.1109/ICACCI.2018.8554842.

[3] H. Poojary, S. B. Shabari, A survey on plant disease detection using support vector machine, in: 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), 2018, pp. 292–295. doi:10.1109/ICCPCCT.2018.8574314.

[4] B. S. Prajapati, V. K. Dabhi, H. B. Prajapati, A survey on detection and classification of cotton leaf diseases, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 2499–2506. doi:10.1109/ICEEOT.2016.7755143.

[5] C. G. Dhaware, K. Wanjale, A modern approach for plant leaf disease classification which depends on leaf image processing, in: 2017 International Conference on Computer Communication and Informatics (ICCCI), IEEE, 2017, pp. 1–4.

[6] J. P. R. Nayak, K. Anitha, B. D. Parameshachari, R. Banu, P. Rashmi, Pcb fault detection using image processing, IOP Conference Series: Materials Science and Engineering 225 (2017) 012244. URL: https://dx.doi.org/10.1088/1757-899X/225/1/012244. doi:10.1088/1757-899X/225/1/012244.

[7] J. D. Pujari, R. Yakkundimath, A. S. Byadgi, Recognition and classification of produce affected by identically looking powdery mildew disease, Acta Technologica Agriculturae 17 (3914) 29–34. URL: https://doi.org/10.2478/ata-2014-0007. doi:doi:10.2478/ata-2014-0007.

[8] J. D. Pujari, R. Yakkundimath, A. S. Byadgi, Image processing based detection of fungal diseases in plants, Procedia Computer Science 46 (2015) 1802–1808. doi:https://doi.org/10.1016/j.procs.2015.02.137.

[9] A. K. Singh, B. Ganapathysubramanian, S. Sarkar, A. Singh, Deep learning for plant stress phenotyping: Trends and future perspectives, Trends in Plant Science 23 (2018) 883–898. URL: https://www.sciencedirect.com/science/article/pii/S1360138518301572. doi:https://doi.org/10.1016/j.tplants.2018.07.004.

[10] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–44. doi:10.1038/nature14539.

[11] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 25, Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[12] M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, A. Moussaoui, Deep learning for plant diseases: Detection and saliency map visualisation, in: Human and Machine Learning, 2018. URL: https://api.semanticscholar.org/CorpusID:52197024.

[13] J. Maurício, I. Domingues, J. Bernardino, Comparing vision transformers and convolutional neural networks for image classification: A literature review, Applied Sciences 13 (2023) 5521.

[14] G. Singh, A critical analysis of convolutional neural networks for leaf disease detection in plants, in: Artificial Intelligence and Smart Agriculture: Technology and Applications, Springer, 2024, pp. 461–493.

[15] M. M. Khalid, O. Karan, Deep learning for plant disease detection, International Journal of Mathematics, Statistics, and Computer Science 2 (2024) 75–84.

[16] M. H. Najim, S. K. Abdulateef, A. H. Alasadi, Early detection of tomato leaf diseases based on deep learning techniques, Int J Artif Intell 13 (2024) 509–515.

[17] V. Singh, N. Sharma, S. Singh, A review of imaging techniques for plant disease detection, Artificial Intelligence in Agriculture 4 (2020) 229–242.

[18] G. Sandhu, R. Kaur, Plant disease detection techniques: A review, 2019, pp. 34–38. doi:10.1109/ICACTM.2019.8776827.

[19] K. Bashir, M. Rehman, M. Bari, Detection and classification of rice diseases: An automated approach using textural features, Mehran University Research Journal of Engineering and Technology 38 (2019) 239–250. URL: https://publications.muet.edu.pk/index.php/muetrj/article/view/759. doi:10.22581/muet1982.1901.20.

[20] A. N. I. Masazhar, M. M. Kamal, Digital image processing technique for palm oil leaf disease detection using multiclass svm classifier, in: 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), 2017, pp. 1–6. doi:10.1109/ICSIMA.2017.8311978.

[21] S. Abed, A. A. Esmaeel, A novel approach to classify and detect bean diseases based on image processing, in: 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2018, pp. 297–302. doi:10.1109/ISCAIE.2018.8405488.

[22] S. Kaur, S. Pandey, S. Goel, Semi-automatic leaf disease detection and classification system for soybean culture, IET Image Processing 12 (2018) 1038–1048.

[23] H. Wang, G. Li, Z. Ma, X. Li, Application of neural networks to image recognition of plant diseases, in: 2012 International Conference on Systems and Informatics (ICSAI2012), 2012, pp. 2159–2164. doi:10.1109/ICSAI.2012.6223479.

[24] S. D. Khirade, A. Patil, Plant disease detection using image processing, in: 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768–771. doi:10.1109/ICCUBEA.2015.153.

[25] S. Zhang, Y. Zhu, Z. You, X. Wu, Fusion of superpixel, expectation maximization and phog for recognizing cucumber diseases, Computers and Electronics in Agriculture 140 (2017) 338–347. URL: https://www.sciencedirect.com/science/article/pii/S0168169917302910. doi:https://doi.org/10.1016/j.compag.2017.06.016.

[26] S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund, P. Kulkarni, "diagnosis and classification of grape leaf diseases using neural networks", in: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1–5. doi:10.1109/ICCCNT.2013.6726616.

[27] A. Rastogi, R. Arora, S. Sharma, Leaf disease detection and grading using computer vision technology & fuzzy logic, in: 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 2015, pp. 500–505. doi:10.1109/SPIN.2015.7095350.

[28] P. M. Mainkar, S. Ghorpade, M. P. Adawadkar, Plant leaf disease detection and classification using image processing techniques, 2015. URL: https://api.semanticscholar.org/CorpusID:212558265.

[29] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, R. Pramodhini, Plant leaf disease detection using computer vision and machine learning algorithms, Global Transitions Proceedings 3 (2022) 305–310.

[30] S. Barburiceanu, S. Meza, B. Orza, R. Malutan, R. Terebes, Convolutional neural networks for texture feature extraction. applications to leaf disease classification in precision agriculture, IEEE Access 9 (2021) 160085–160103. doi:10.1109/ACCESS.2021.3131002.

[31] V. Singh, A. Misra, Detection of plant leaf diseases using image segmentation and soft computing techniques, Information Processing in Agriculture 4 (2017) 41–49. URL: https://www.sciencedirect.com/science/article/pii/S2214317316300154. doi:https://doi.org/10.1016/j.inpa.2016.10.005.

[32] S. Ramesh Maniyath, V. P V, N. M, P. R, P. N, S. N, H. Ram, Plant disease detection using machine learning, 2018, pp. 41–45. doi:10.1109/ICDI3C.2018.00017.

[33] J. G. Arnal Barbedo, Plant disease identification from individual lesions and spots using deep learning, Biosystems Engineering 180 (2019) 96–107. URL: https://www.sciencedirect.com/science/article/pii/S1537511018307797. doi:https://doi.org/10.1016/j.biosystemseng.2019.02.002.

[34] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, D. Traore, Deep neural networks with transfer learning in millet crop images, Computers in Industry 108 (2019) 115–120. URL: https://www.sciencedirect.com/science/article/pii/S0166361518305888. doi:https://doi.org/10.1016/j.compind.2019.02.003.

[35] A. Fuentes, S. Yoon, S. C. Kim, D. S. Park, A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition, Sensors 17 (2017). URL: https://www.mdpi.com/1424-8220/17/9/2022. doi:10.3390/s17092022.

[36] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, B. Tekinerdogan, Analysis of transfer learning for deep neural network based plant classification models, Computers and Electronics in Agriculture 158 (2019) 20–29. URL: https://www.sciencedirect.com/science/article/pii/S0168169918315308. doi:https://doi.org/10.1016/j.compag.2019.01.041.

[37] K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Computers and Electronics in Agriculture 145 (2018) 311–318. URL: https://www.sciencedirect.com/science/article/pii/S0168169917311742. doi:https://doi.org/10.1016/j.compag.2018.01.009.

[38] K. Roy, S. Chaudhuri, J. Frnda, S. Bandopadhyay, I. Ray, S. Banerjee, J. Nedoma, Detection of tomato leaf diseases for agro-based industries using novel pca deepnet, IEEE Access PP (2023) 1–1. doi:10.1109/ACCESS.2023.3244499.

[39] M. Fawakherji, A. Youssef, D. Bloisi, A. Pretto, D. Nardi, Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation, in: 2019 Third IEEE International Conference on Robotic Computing (IRC), 2019, pp. 146–152. doi:10.1109/IRC.2019.00029.

[40] X. Zhang, Y. Qiao, F. Meng, C. Fan, M. Zhang, Identification of maize leaf diseases using improved deep convolutional neural networks, IEEE Access 6 (2018) 30370–30377. doi:10.1109/ACCESS.2018.2844405.

[41] S. C. K., J. C. D, N. Patil, Cardamom plant disease detection approach using efficientnetv2, IEEE Access 10 (2022) 789–804.

[42] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan, A. A. Khan, Detection of apple plant diseases using leaf images through convolutional neural network, IEEE Access 11 (2023) 6594–6609. doi:10.1109/ACCESS.2022.3232917.

[43] P. Eddy, A. Smith, B. Hill, D. Peddle, C. Coburn, R. Blackshaw, Comparison of neural network and maximum likelihood high resolution image classification for weed detection in crops: Applications in precision agriculture, 2006, pp. 116 – 119. doi:10.1109/IGARSS.2006.35.

[44] S. P. Mohanty, D. P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Frontiers in Plant Science 7 (2016). URL: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419. doi:10.3389/fpls.2016.01419.

[45] A. M. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. S. Al-Rakhami, A. H. Gumaei, Automatic detection of citrus fruit and leaves diseases using deep neural network model, IEEE Access 9 (2021) 112942–112954. URL: https://api.semanticscholar.org/CorpusID:236473673.

[46] S. Hassan, A. Maji, Plant disease identification using a novel convolutional neural network, IEEE Access PP (2022) 1–1. doi:10.1109/ACCESS.2022.3141371.

[47] C. Zhou, S. Zhou, J. Xing, J. Song, Tomato leaf disease identification by restructured deep residual dense network, IEEE Access PP (2021) 1–1. doi:10.1109/ACCESS.2021.3058947.

[48] T. Sharma, Plant village dataset (updated), 2024. URL: https://www.kaggle.com/datasets/tushar5harma/plant-village-dataset-updated.

[49] G. G., A. P. J., Identification of plant leaf diseases using a nine-layer deep convolutional neural network, Computers & Electrical Engineering 76 (2019) 323–338. URL: https://www.sciencedirect.com/science/article/pii/S0045790619300023. doi:https://doi.org/10.1016/j.compeleceng.2019.04.011.

[50] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Xie, J. Feng, Deepvit: Towards deeper vision transformer, arXiv preprint arXiv:2103.11886 (2021).

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[53] M. K. Siddiqui, X. Huang, R. Morales-Menendez, N. Hussain, K. Khatoon, Machine learning based novel cost-sensitive seizure detection classifier for imbalanced EEG data sets, International Journal on Interactive Design and Manufacturing (IJIDeM) 14 (2020) 1491–1509.

[54] M. K. Siddiqui, R. Morales-Menendez, X. Huang, N. Hussain, A review of epileptic seizure detection using machine learning classifiers, Brain informatics 7 (2020) 1–18.

[55] D. L. Streiner, J. Cairney, What's under the roc? an introduction to receiver operating characteristics curves, The Canadian Journal of Psychiatry 52 (2007) 121–128.

[56] A. G. Lalkhen, A. McCluskey, Clinical tests: sensitivity and specificity, Continuing Education in Anaesthesia Critical Care & Pain 8 (2008) 221–223.

[57] N. A. Obuchowski, Receiver operating characteristic curves and their use in radiology, Radiology 229 (2003) 3–8.

[58] M. K. Siddiqui, R. Morales-Menendez, S. Ahmad, Application of receiver operating characteristics (ROC) on the prediction of obesity, Brazilian Archives of Biology and Technology 63 (2020). doi:https://doi.org/10.1590/1678-4324-2020190736.

[59] A. A. Aljumah, M. K. Siddiqui, M. G. Ahamad, Application of classification based data mining technique in diabetes care, Journal of Applied Sciences 13 (2013) 416–422. doi:10.3923/jas.2013.416.422.

[60] S.-T. Yeh, et al., Using trapezoidal rule for the area under a curve calculation, Proceedings of the 27th Annual SAS® User Group International (SUGI'02) (2002).

[61] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Neural Information Processing Systems, 2017. URL: https://api.semanticscholar.org/CorpusID:21889700.

[62] S. Hossain, M. Tanzim Reza, A. Chakrabarty, Y. J. Jung, Aggregating different scales of attention on feature variants for tomato leaf disease diagnosis from image data: A transformer driven study, Sensors 23 (2023) 3751.

[63] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, S. G. Koolagudi, Tomato leaf disease detection using convolutional neural networks, in: 2018 Eleventh International Conference on Contemporary Computing (IC3), 2018, pp. 1–5. doi:10.1109/IC3.2018.8530532.

[64] R. A. Boukabouya, A. Moussaoui, M. Berrimi, Vision transformer based models for plant disease detection and diagnosis, in: 2022 5th International Symposium on Informatics and its Applications (ISIA), IEEE, 2022, pp. 1–6.

[65] U. Barman, P. Sarma, M. Rahman, V. Deka, S. Lahkar, V. Sharma, M. J. Saikia, Vit-smartagri: Vision transformer and smartphone-based plant disease detection for smart agriculture, Agronomy 14 (2024) 327.

[66] K. Shaheed, I. Qureshi, F. Abbas, S. Jabbar, Q. Abbas, H. Ahmad, M. Z. Sajid, Efficientrmt-net—an efficient resnet-50 and vision transformers approach for classifying potato plant leaf diseases, Sensors 23 (2023) 9516.

[67] U. Barman, D. Sahu, G. G. Barman, J. Das, Comparative assessment of deep learning to detect the leaf diseases of potato based on data augmentation, in: 2020 International Conference on Computational Performance Evaluation (ComPE), 2020, pp. 682–687. doi:10.1109/ComPE49325.2020.9200015.

[68] W. Chen, J. Chen, A. Zeb, S. Yang, D. Zhang, Mobile convolution neural network for the recognition of potato leaf disease images, Multimedia Tools and Applications 81 (2022) 20797–20816.

[69] D. W. Hosmer Jr, S. Lemeshow, R. X. Sturdivant, Applied logistic regression, volume 398, John Wiley & Sons, 2013.