
# Plant Disease Detection

EE 593 - DDP Stage I

Ayush Munjal, 19D070014
Department of Electrical Engineering
Indian Institute of Technology Bombay, Mumbai

Guided By Prof. Rajbabu Velmurugan

# Abstract

The document discusses the task of plant disease detection and classification, with a specific focus on tomato and onion plants. We explore the use of transformers for this task and compare their performance with CNNs.

# Contents

# 1 Introduction

Detecting plant diseases is vital for agriculture and food security, especially in countries like India, where agriculture is a major industry. It helps prevent crop losses, economic damage to farmers, and food shortages. Timely disease detection enables targeted treatments, reducing the need for chemical interventions and promoting sustainable farming practices. In essence, it plays a crucial role in ensuring a reliable food supply in a growing world.

We start with working on the PlantVillage dataset then we extend our work to onion plant disease images taken from the field. The Plant Village dataset is a rich resource for agricultural research and computer vision. It includes high-resolution images of plants affected by various diseases and pests, making it invaluable for training machine-learning models like CNNs. Researchers use this dataset to develop accurate disease detection models, benefiting precision agriculture and promoting sustainable farming practices. There is a large difference in the quality of images of both data as well as the number of images which will be seen later.

We begin by employing a basic CNN model, DenseNet121, achieving satisfactory accuracy figures. We then delve into the model's qualitative performance through Grad-CAM and t-SNE analysis. Following this, we adopt two transformer models, ViT (Vision Transformer) and DeiT (Data-Efficient Image Transformer), to compare their results against the DenseNet121 model. Additionally, we explore the concept of background removal. Lastly, we apply the transformer model to the onion field data and evaluate its performance.

# 2 Literature survey

## 2.1 An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Vision Transformers (ViTs) are a type of neural network architecture that has achieved state-of-the-art results on a variety of image classification benchmarks. ViTs are based on the transformer architecture, which was originally developed for natural language processing (NLP).

ViTs work by first dividing an image into a grid of patches. Each patch is then embedded into a high-dimensional vector using a learnable embedding function. The embedded patches are then fed into the transformer encoder, which learns to extract long-range dependencies in the image. The output of the transformer encoder is then fed to a linear classifier, which predicts the class of the image.

ViTs have a number of advantages over traditional convolutional neural networks (CNNs) for image classification tasks. First, ViTs are able to learn long-range dependencies in images without relying on hand-crafted features. Second, ViTs are more efficient to train than CNNs, especially on large datasets. One challenge with ViTs is that they can be computationally expensive to train and deploy. This is because ViTs require a large amount of memory to store the attention matrices.

## 2.2 Training data-efficient image transformers and distillation through attention

Vision transformers (ViTs) have achieved state-of-the-art results on a variety of image classification benchmarks. However, ViTs are typically trained on very large datasets, such as ImageNet, which can be expensive and time-consuming to collect and label. In the paper "Training data-efficient image transformers & distillation through attention", the authors propose a new method for training data-efficient ViTs using distillation through attention. Distillation is a technique for transferring knowledge from a large, pre-trained model to a smaller, student model.

The authors' method works by first pre-training a large ViT (teacher model) on a large dataset. The teacher model is then used to generate distillation tokens for each training image. These distillation tokens encode the teacher model's knowledge about the image in a condensed form. The student model is then trained to predict the distillation tokens, rather than the class labels of the images. This training strategy forces the student model to learn the same features as the teacher model, but without the need for a large dataset.

The authors evaluated their proposed method on a variety of image classification benchmarks, including ImageNet and CIFAR-10. They found that their method was able to train data-efficient ViTs that achieved state-of-the-art results.

## 2.3 Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Gradient-Weighted Class Activation Mapping (Grad-CAM) is a technique for visualizing the important regions in an image for a particular class prediction. Grad-CAM is based on the idea that the gradients of the class score with respect to the input image can be used to identify the regions that are most important for the prediction.

Grad-CAM works by first computing the gradients of the class score with respect to the input image. These gradients are then weighted by the global average pooling of the feature maps from the last convolutional layer. The weighted gradients are then used to create a heatmap, which highlights the important regions in the image for the class prediction. The Grad-CAM heatmap can be overlaid on the original image to produce a visualization of the important regions. This visualization can be used to understand why the model made the particular prediction and to identify any potential biases in the model.

## 2.4 Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features

The paper "Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features" by Radhakrishnan et al. (2022) proposes a new framework for feature learning that is based on the average gradient outer product (AGOP). The AGOP is a measure of how different features interact with each other to predict the output of a model.

NFM(ith layer Neural feature matrix) is proportional to average gradient outer product of the network with respect to input to this layer. Deep Neural Feature Ansatz is that deep neural network identify important features by ranking them on basis of magnitude of change in prediction upon perturbation. RFM(Recursive feature machine) proposed which uses a (learnable) kernel based predictor then uses NFM to learn features using outer gradient product. Features learnt from RFM have been found to have high correlation with features learned using DNNs. RFM achieved SOTA performance in classification and regression tasks in less iterations.

# 3 PlantVillage Data

The Plant Village dataset is a valuable resource that contains an extensive collection of plant images, complete with detailed annotations and metadata. Originating from a diverse range of locations and climates, this dataset comprises over 50,000 high-resolution images that capture various plant species affected by approximately 38 different diseases and pests. Compiled over several years, this dataset represents a comprehensive and global effort to document plant health issues. The Plant Village dataset has played a pivotal role in training and validating machine learning models, with its substantial size and diversity offering a wealth of opportunities for research in precision agriculture.

It contains 54,303 images for 38 different plant diseases. In our work, we focus on tomato plants, which are represented by a subset of 18,160 images covering nine diseases and one healthy class. The data is quite imbalanced as shown in the fig2. Some sample images are shown below.

Figure 1: Sample images from PlantVillage dataset

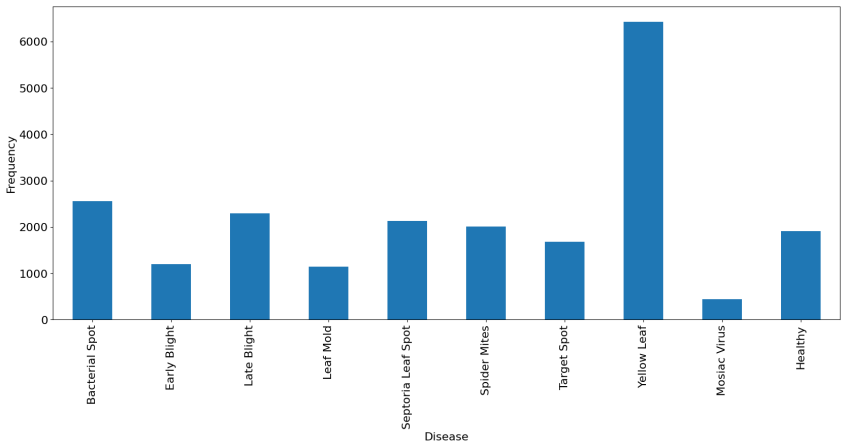


Figure 2: Histogram of data of different classes of tomato plant

# 4 Experiment using PlantVillage dataset

## 4.1 DenseNet121 model

DenseNet121 is a deep convolutional neural network model renowned for its exceptional performance in image classification tasks. It stands out for its dense connectivity pattern, where each layer receives input from all previous layers, promoting feature reuse and reducing the number of parameters. This results in both efficient training and impressive accuracy, making DenseNet121 a popular choice in computer vision applications. Its architecture is as shown in the fig3. It consists of 4 dense blocks, each block containing different number of dense layers. We use the DenseNet121 pre-trained on the ImageNet model then we fine-tune this model on the PlantVillage dataset.

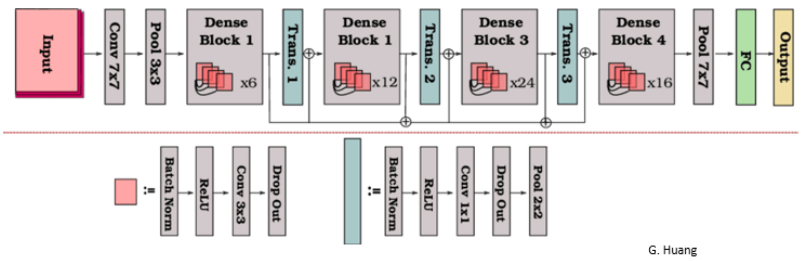


Figure 3: DenseNet121 architecture

We divide the data into 80% training and 20% validation data. DenseNet121 model is trained on this data for 20 epochs. From this, we get the best validation accuracy of 92.5%.

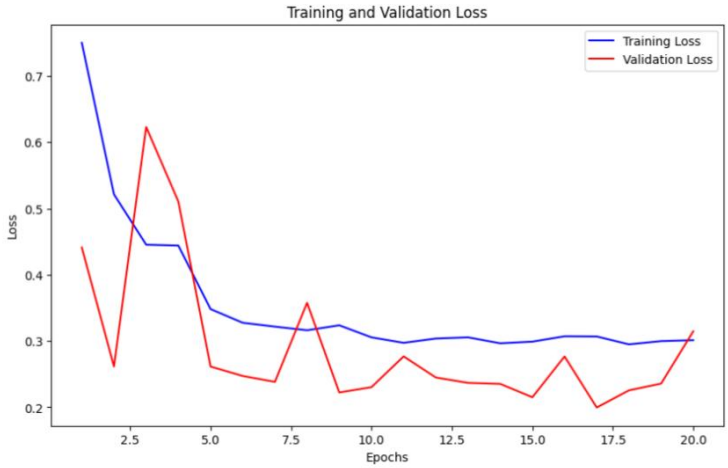


Figure 4: DenseNet121 training and validation loss curve

We also use GradCAM to visualize activation maps to get a better understanding of the model. GradCAM, short for Gradient-weighted Class Activation Mapping, is a technique used to visualize and understand the regions in an image that contribute the most to a deep neural network's classification decision. It leverages the gradient information of the network's final classification score with respect to the feature maps in the final convolutional layer. By multiplying these gradients and the feature maps, GradCAM assigns importance scores to each spatial location in the feature maps. These scores represent how crucial each part of the image is for the network's decision. GradCAM then combines these importance scores to produce a heatmap, highlighting the regions that strongly influence the classification result. This visualization method aids in interpreting and validating the decisions made by deep learning models, offering insights into their decision-making process. From the results above we can see that our model doesn't

Figure 5: Original images

specifically focus on the diseased part in any of the images. So, although we are getting good accuracy metrics our model is not focusing on the important part of images to improve this we try transformers as discussed further.

We also perform t-SNE analysis for the model on different dense blocks to understand how much important each block is. t-Distributed Stochastic Neighbor Embedding is a technique for dimensionality reduction for visualization of high-dimension data. It is used to explore patterns within the data. The t-SNE results for different dense blocks and the final model are as shown below. We can observe from the above result that as we go

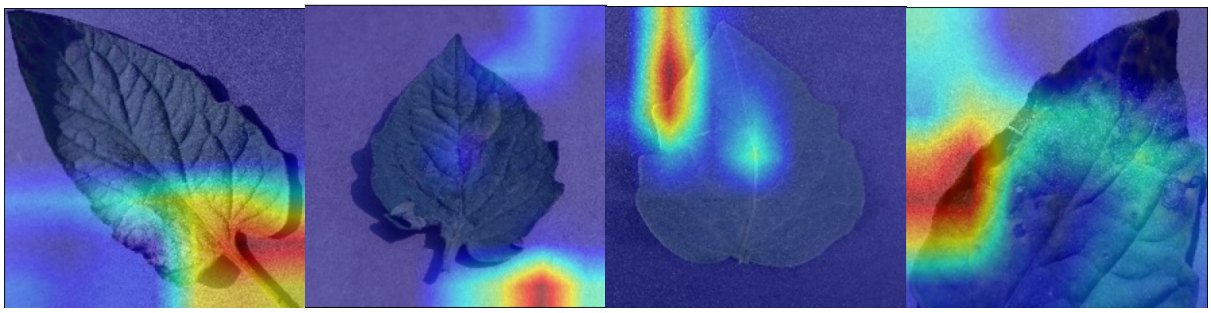

Figure 6: GradCAM results

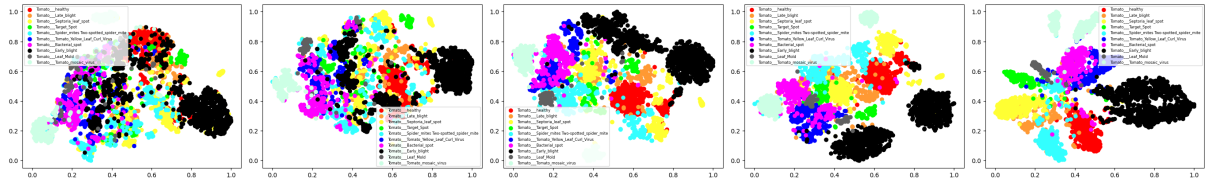

Figure 7: t-SNE plots of Denseblock 1, 2, 3, 4 & final model

latter layers the clusters in t-SNE plots become more separated as expected, which shows that the latter layers contribute more in predicting and separating the data of different classes.

## 4.2 Transformers & Attention

Using DenseNet we got a good accuracy value but our model was not focusing on important parts of images so to improve this we try transformers as we expect attention to improve this. Transformers have revolutionized the field of deep learning, particularly in the realm of natural language processing and computer vision. At the heart of a transformer's architecture is the concept of attention, which plays a pivotal role in its impressive performance. Attention mechanisms allow the model to focus on specific parts of input sequences, assigning varying levels of importance to different elements. This dynamic and adaptive approach to feature extraction and context understanding has significantly improved the ability of transformers to handle sequences of varying lengths and complexities. Attention mechanisms enable transformers to capture long-range dependencies and relationships within data, making them exceptionally well-suited for tasks like language translation, sentiment analysis, and image recognition. The ability to attend to relevant information across the entire input sequence sets transformers apart, contributing to their widespread adoption and remarkable success in various machine-learning applications. Transformers use different kinds of attention based on their architecture self-attention, cross-attention, etc. For now, we will be working with self-attention only. One issue with transformers is that they require a very large amount of data for training which will be an issue for us as our second dataset (onion plant) is very limited. We look at two very popular transformers Vision Transformer (ViT) and Data Efficient Image Transformer (DeiT).

### 4.2.1 Vision Transformer (ViT)

ViT is a groundbreaking deep learning architecture that extends the power of transformers to the domain of computer vision. Originally designed for natural language processing, transformers have been adapted to process and analyze visual data, making ViT an influential model in image recognition tasks. Instead of using convolutional layers, ViT relies on a self-attention mechanism to capture complex spatial relationships within images. This approach allows ViT to effectively model global context and relationships among image pixels, leading to impressive results in tasks like object detection, image classification, and segmentation. The Vision Transformer has significantly advanced the field of computer vision and continues to be a driving force in the development of innovative visual recognition systems. The architecture of the ViT model is as shown in the fig8. We use the ViT model pre-trained on ImageNet 21k. Different variants of the ViT model are available based on patch size, architecture, and pre-training dataset.

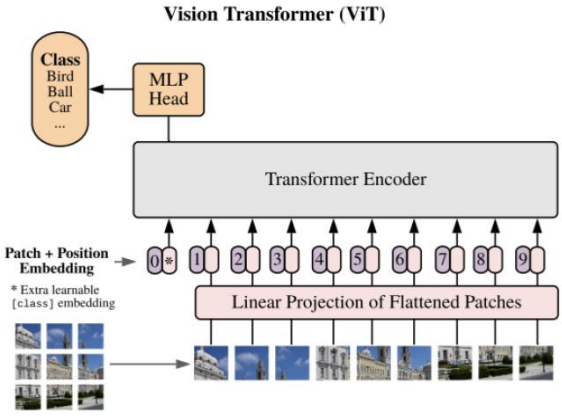


Figure 8: Vision Transformer architecture

### 4.2.2 Data Efficient Image Transformer (DeiT)

DeiT, or Data-efficient Image Transformer, is a notable advancement in computer vision. It extends the power of transformer models, originally designed for natural language processing, to image classification tasks. DeiT leverages a teacher-student training paradigm, where a large-scale convolutional neural network (CNN) serves as the teacher to guide a smaller transformer-based student model. This approach allows DeiT to achieve remarkable results with fewer labeled training examples, significantly enhancing data efficiency in image classification tasks. DeiT's success showcases the potential for transformer architectures in computer vision and opens new possibilities for tackling image-related challenges with greater efficiency and performance. The architecture of DeiT is as show in the fig9. The main advantage of DeiT over other transformers is that it provides almost the same performance with much less computation cost. We use DeiT pre-trained on ImageNet-1k.

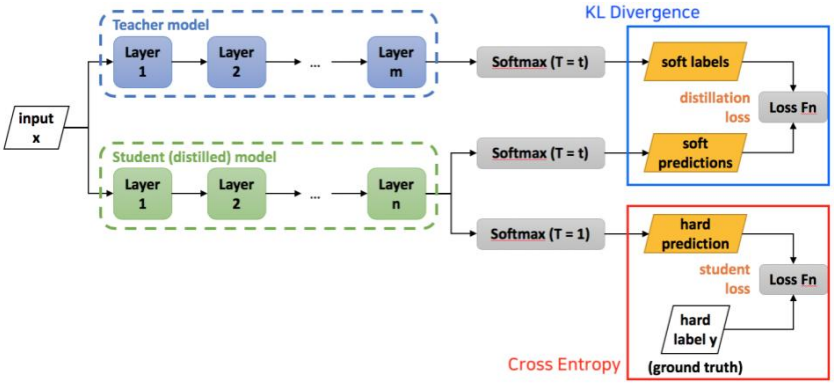


Figure 9: Data Efficient Image Transformer architecture

### 4.2.3 Results

ViT and DeiT were fine-tuned on PlantVillage tomato data with an 80-20 split. Using ViT we got test accuracy of 98.8% and 95% using DeiT. A comparison of these models with the DenseNet121 model is shown in the table below.

| | DenseNet121 | DeiT | ViT |
|---|---|---|---|
| Best train accuracy | 89% (20 epochs) | 91% (20 epochs) | 98% (10 epochs) |
| Test accuracy | 92% | 95% | 98% |
| Time for training(s/epoch) | 77 | 60 | 450 |
| Size of model(MB) | 29 | 21.4 | 327.3 |

As we can see from the above table both transformers outperform the DenseNet model. Especially ViT model gives better performance than DenseNet and DeiT in 10 epochs only with the other two being trained for 20 epochs. But there is one issue with the ViT model its model size and training time are much larger than the DeiT model, the size of ViT is more than 15x of DeiT, training time of ViT is 7x times more than DeiT. So, even though the ViT model provides better accuracy but this comes at the cost of heavy computation due to this we decided to move with the DeiT model as our final goal is to employ our model on an ML chip so we want our model size to be small and DeiT provides us good performance with much less computation cost even less than DenseNet. So from now on we will work with the DeiT model only. The training for all these models was performed on google colaboratory.

### 4.2.4 Attention maps for DeiT model

Our primary goal was to improve our model not only in accuracy but to also increase the interpretability of our model so that it recognizes important parts of images and focuses on them only, now we will evaluate this on our transformer models. We

plot attention maps for DeiT. Attention maps in transformers play a fundamental role in understanding how these models process input data. These maps depict the strength of connections or attention between different parts of the input sequence or image. When applied to natural language tasks, they illustrate which words or tokens are most influential in predicting the next word. In computer vision, attention maps highlight which regions of the input image the model "attends" to when making predictions. This visualization is crucial for transparency, interpretability, and debugging in transformer models, allowing researchers and practitioners to gain insights into how the model processes information and forms its predictions. It serves as a powerful tool for understanding the inner workings of transformers, making them not just effective but also more interpretable. Shown below are the attention maps for the DeiT model.

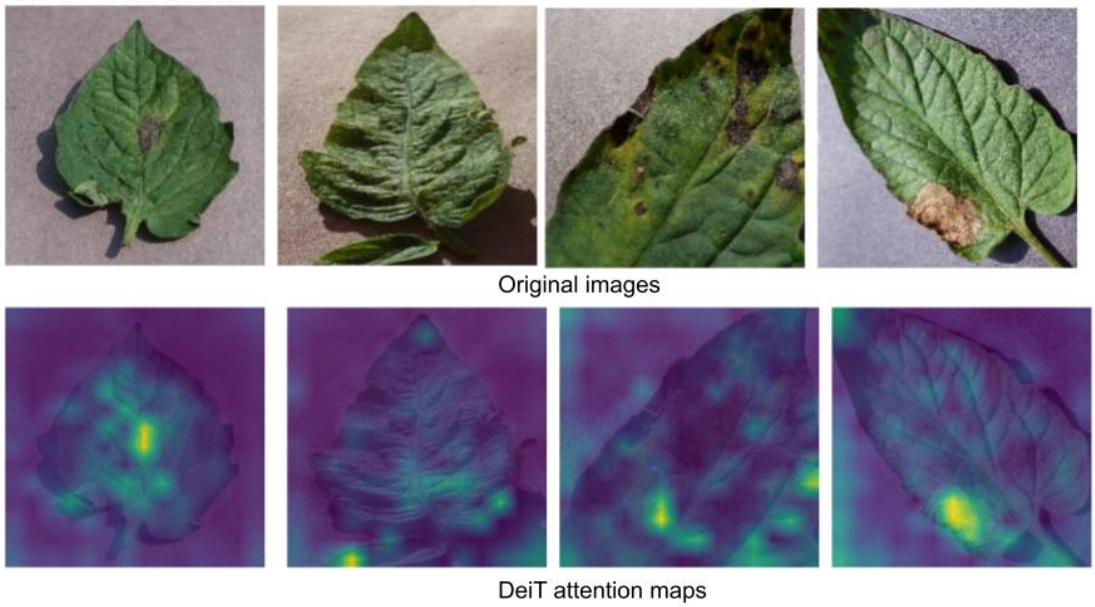


Figure 10: Attention maps for DeiT model

We can observe that these attention maps are much better than what we got for DenseNet as in the first and fourth images almost full focus is on the diseased part for the second image and third image some diseased parts are ignored but still better than what we got for DenseNet model.

# 5 Onion Dataset

Now, we have seen that DeiT gives us much better results than the DenseNet model in terms of interpretability so now we will use this model on our original problem, i.e., to train a model for onion plant images taken from the field.

## 5.1 Dataset

This dataset consists of six classes with 2,133 total images for six classes. It contains images of two types of onion plants. Number of images for each disease is as shown below:

- Bulb rot: 717 images
- Healthy Bulb: 17 images
- Healthy seed: 99 images
- IYSV: 754 images
- Thrips Insect: 240 images
- Thrips Symptoms: 306 images

We work with Bulb Rot, IYSV, and Thrips (insect and symptoms combined) classification for now due to data constraints. Some sample images are shown below

Figure 11: Sample images from the onion dataset

## 5.2 Results

We train the DeiT model pre-trained on ImageNet on this data for 10 epochs with a 50-25-25 data split. After training we got 100% training accuracy, 99.6% validation accuracy, and 99.8% testing accuracy. From these values, it is clear that the model is overfitting the data, which became more clear after plotting the confusion matrix for validation and testing data.

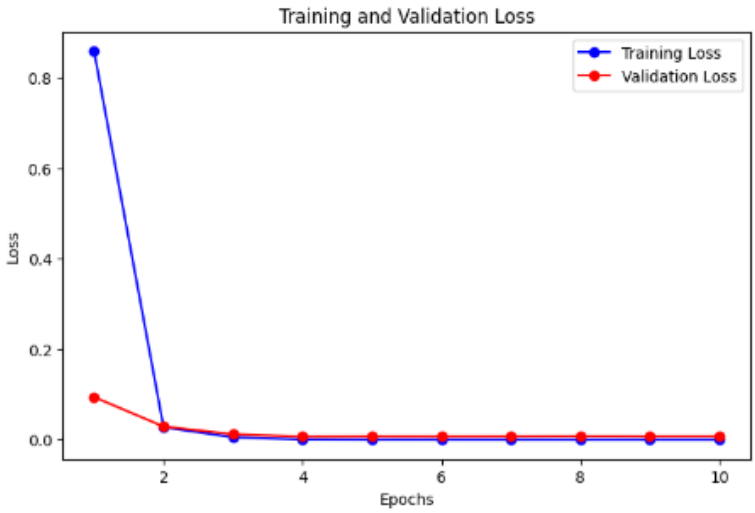


Figure 12: Training and validation loss

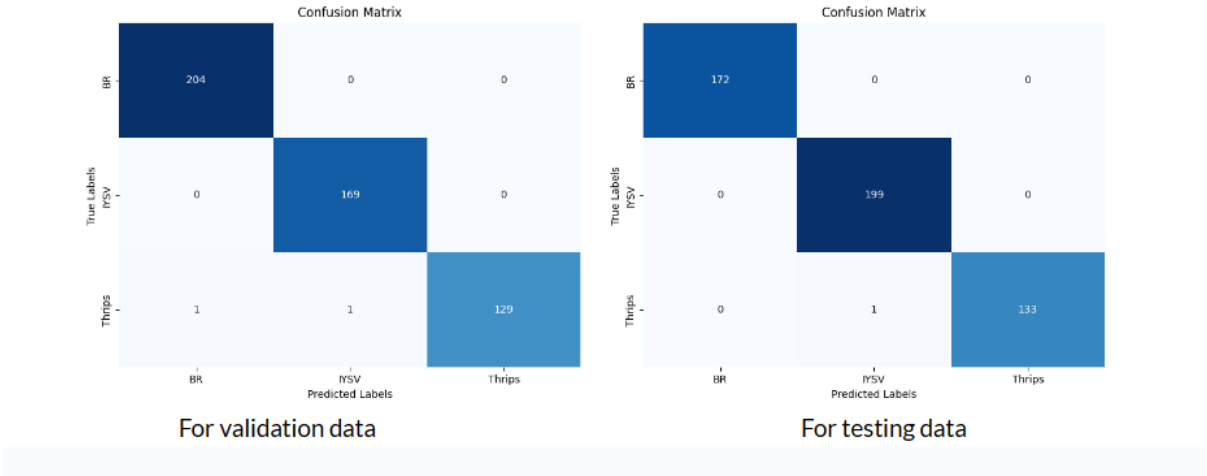


Figure 13: Confusion matrix for validation and testing data

There are two reasons why there is overfitting, first for transformers we require a large amount of data but in this case, we have only around 2000 images which is much less for training a transformer. Also, there are many repetitions in data this is because photos of plants are taken each day so there are very minor changes in that case also photos of a single plant are taken from different angles which also leads to data repetition. Due to this, our effective data is very less due to which are having overfitting issues.

The attention maps for some correctly classified images are as shown below

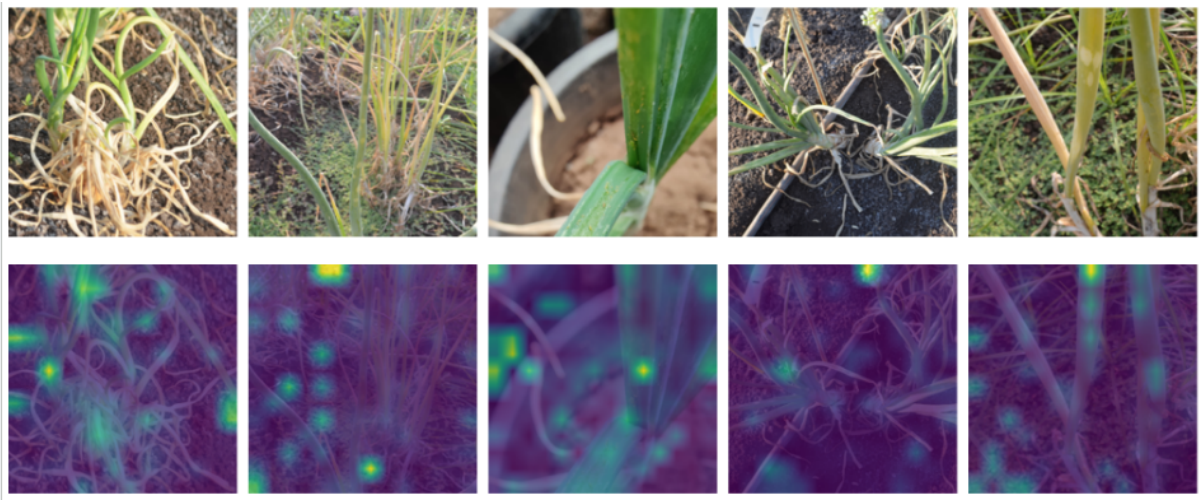

Figure 14: Attention maps for DeiT model

We can see that the model is focusing on some of the important parts of images in some cases but the background is interfering much with the plant. So, although the model is overfitting attention maps in some of the cases are good.

# 6 Conclusion

We saw that in some cases when we are getting good accuracy values the model interpretability is not as good as it was the case with DenseNet, so we tried transformers and got much better results not only in terms of accuracy metrics but also model interpretability.

We then tried the DeiT model on the onion dataset but we faced the issue of overfitting. Two main challenges we need to address for the onion dataset are training the transformer model using less data and background suppression although the background suppression for this case will not be easy because of type of the images we have.

# References


[1] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 618-626)

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Uszkoreit, J., Unterthiner, T., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[3] Radhakrishnan, A., Beaglehole, D., Pandit, P., & Belkin, M. (2022). Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features. arXiv preprint arXiv:2212.13881.

[4] Touvron, H., Cord, M., Douze, M., Jegou, H., Joulin, A., & Lapoujade, M. (2021). Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877.

[5] Nantheera Anantrasirichai, Sion Hannuna, and Nishan Canagarajah. "Automatic Leaf Extraction from Outdoor Images." arXiv preprint arXiv:1709.06437 (2017).