An Agriculture Experiment: Applied Vision Transformer for Mango Leaf Disease Detection

Tran Tien Thanh*
FPT University, HCM
Phone: 0901885321
Email: tienthanhpro2003@gmail.com

Nguyen Luong Tuan
FPT University, HCM
Phone: 0984582105
Email: nguyenluongtuan1309@gmail.com

Vo Minh Huy
FPT University, HCM
Phone: 0961034458
Email: vomhuy051204@gmail.com

***Abstract*—This paper introduces the implementation of Vision Transformer (ViT), a computer vision model based on some elements of Transformer to perform the usage of image classification, which in this case is to predict and detect the mango tree diseases. Although the use of this technology can be considered superfluous, when for some long-time farmers, with the eye they can already know the type of disease on the crop. However, the problem that we are concerned about is that if with large crops, looking at each leaf is not really effective and causes delays in early detection and prevention. Our approach is not only fast but also highly accurate, so that appropriate prevention measures can be taken, preventing unintended cases. Although the initial results had certain difficulties when implementing such as long running time, RAM problems, low accuracy in the first process, but after many experiments, we finally found solutions and gave very positive results in the next training times. The result are very positive with 96% accuracy, a crucial step forward compared to other CNN model such as VGG-16 and efficentNet. We hope that this project can contribute to the country's long-standing agriculture, so that the boundary between technology and farming life is not an obstacle to development.**

***Keywords*—** vision transformer, image classification

I. Introduction

In everyday life, the need for food is undeniable and agriculture plays an extremely important role in meeting this demand. Especially, in Vietnam – a country located in the rich and strong Mekong Delta region in agriculture, the agricultural sector is not only a source of food supply but also a pillar of the economy. However, to ensure the yield and quality of agricultural products, caring for and protecting crops from diseases and pests is no small challenge. In the context of climate change and the rapid spread of new diseases, accurate and timely diagnosis of diseases has become essential, and the application of technology in this field has become even more urgent.

However, practice shows that diagnosing diseases of crops is not a simple task. The manifestations of the disease are often complex and difficult to detect, making it challenging to accurately identify the type of disease and apply treatment. In particular, in the case of mango trees, the symptoms of common diseases such as leaf spot, root rot or mold are often quite vague and easily confused with other problems such as lack of nutrition, pollution or weather.Similarly, diagnosing diseases across an entire tree with numerous leaves, along with the potential presence of multiple diseases, can lead to inaccuracies in diagnosis or even failures in prediction. Therefore, the demand for accurate and effective methods of disease diagnosis in agriculture is increasing, while creating great opportunities for the application of technology and artificial intelligence in this field.

To address this challenge, we propose an innovative and cutting-edge solution: using Vision Transformer (ViT) – one of the latest technologies in the field of artificial intelligence. ViT is a deep learning model specifically designed to process image data, with the ability to ”see” and ”understand” through the analysis of important features in images. By applying this to the diagnosis of mango tree disease through leaves, we hope to be able to create a robust diagnostic system, capable of detecting and classifying diseases accurately and quickly. The model's flexibility and self-learning ability are also important factors that help minimize dependence on specific rules and characteristics of each disease, creating a general and effective solution for many situations in real agriculture.

The results of this study are an important step forward in solving the problem of diagnosing diseases of mango trees in agriculture. ViT has demonstrated outstanding capabilities in classifying various diseases on leaves with high accuracy and significant reliability, but only when tested on individual leaves. Through the testing process using images of each leaf, we have documented a clear level of effectiveness of the model, with a high accuracy rate in diagnosing the disease, while minimizing confusion and enhancing prevention and treatment. This opens the door to the application of technology in agriculture in an efficient and sustainable way, helping to improve yields and product quality, while minimizing losses and costs in crop management and care. This is not only an important step forward in the field of agriculture, but also a testament to the power of technology and artificial intelligence in solving the challenges of modern society.

The rest will be outlined in the article and arranged accordingly. In section II, we will discuss the application and other related works of Vision Transformer models. In section III, we will provide an overview of diagnosing diseases in mango trees as well as describe the vision transformer methodology. Section IV includes descriptions of the data, parameter explanation, our training process, evaluation metrics and our results including comparison of other model. And then in the final concluding remarks, Section V will summarize our findings. Section VI is our sincere gratitude to some of the person who have supported us.

II. Related works

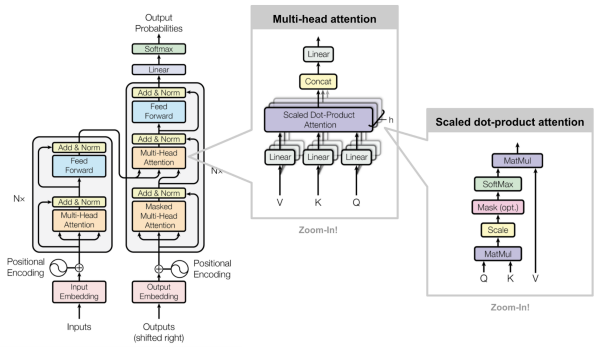

Figure 1. Original Transformer model[1]

ViT is strongly applied in many different aspects, itself a Transformer[1] application adapted to suit Computer Vision. There are four interesting applications of ViT that we will mention here: image classification, image captioning, image partitioning and anomaly detection.

A. Image Classification

Global Context Vision Transformer (GC ViT) is a novel architecture that enhances parameter and compute utilization for computer vision. The method leverages global context self-attention modules, joint with standard local self-attention, to effectively and efficiently model both long and short-range spatial interactions, without the need for expensive operations such as computing attention masks or shifting local windows. The model achieves state-of-the-art results across image classification, object detection and semantic segmentation tasks. On ImageNet-1K dataset for classification, the variants of GC ViT with 51M, 90M and 201M parameters achieve 84.3%, 85.0% and 85.7% Top-1 accuracy, respectively, at 224 image resolution and without any pre-training.[2]

B. Image Partitioning

Dense vision transformers is an architecture that leverages vision transformers in place of convolutional networks as a backbone for dense prediction tasks. The works assemble tokens from various stages of the vision transformer into image-like representations at various resolutions and progressively combine them into full-resolution predictions using a

convolutional decoder. The transformer backbone processes representations at a constant and relatively high resolution and has a global receptive field at every stage. These properties allow the dense vision transformer to provide finer-grained and more globally coherent predictions when compared to fully-convolutional networks. When applied to semantic segmentation, dense vision transformers claimed to have set a new state-of-the-art on ADE20K dataset with 49.02% mIoU.[3]

C. Image Captioning

Image transformer are consist of a modified encoding transformer and an implicit decoding transformer, motivated by the relative spatial relationship between image regions. The design widen the original transformer layer's inner architecture to adapt to the structure of images. With only regions feature as inputs, The paper claimed that the model have achieved new state-of-the-art performance on both MSCOCO offline and online testing benchmarks.[4]

D. Anomaly detection

A transformer-based image anomaly detection and localization network is introduced as a combination of a reconstruction-based approach and patch embedding. The use of transformer networks helps preserving the spatial information of the embedded patches, which is later processed by a Gaussian mixture density network to localize the anomalous areas.[5]

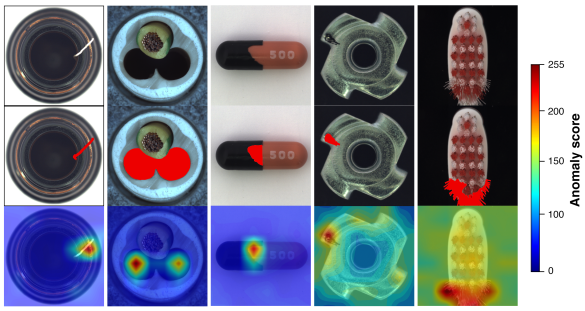

Figure 2. Anomaly detection application[5]

III. Methods

A. Problem Overview

The task at hand involves diagnosing diseases in mango trees based on images. Identifying disease symptoms in mango trees presents a unique challenge due to the multitude of disease types and the often subtle similarities among symptoms. Our objective is to develop an automated system capable of accurately and effectively recognizing and classifying common diseases afflicting mango trees.

B. Model Description

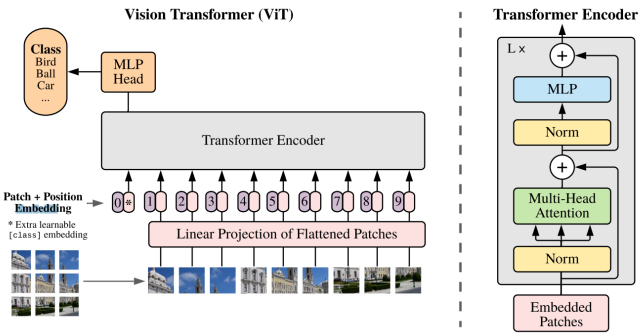

Figure 3. Model structure[6]

Our proposed solution involves utilizing the Vision Transformer (ViT) as the primary model for diagnosing mango tree diseases. ViT is a model employing a transformer architecture originally devised for image processing. It accomplishes this by segmenting an image into smaller components and applying transformer mechanisms to each segment.

The method begins by dividing the input image into smaller parts, called patches, and then applying Linear Projection to transform them into embedding vectors corresponding to each patch. Here, Linear Projection is a Dense layer, using the formula:

$$z_i = W \cdot x_i + b$$

Here:

- x_i : Represents the flattened vector of patch i. In the Vision Transformer model, a patch is represented as a vector by flattening the pixel values of that patch.
- z_i: Corresponds to the output of xi after passing through the Linear Projection process. This means z_i is the result of transforming xi through a linear operation.
- W : The embedding matrix. This matrix consists of weights used in the Linear Projection process to transform vector x_i. Each row of this matrix corresponds to an element in the output vector z_i, and each column corresponds to an element in the input vector x_i.
- b: The bias vector added after performing the linear operation. This bias helps adjust and shift the data after the linear transformation process.

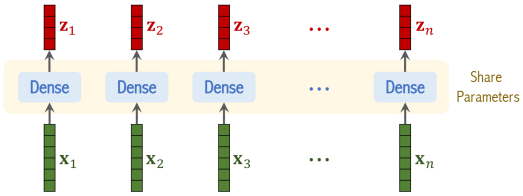

Figure 4. Dense[6], [7], [8]

This process aids in representing information from different parts of the image as vectors of lower-dimensional data. To incorporate positional information for each patch in an image, we adopt a similar approach to the original Transformers model by applying Positional Embedding. After obtaining positional embedding vectors for each patch, we add these vectors correspondingly to the embedding vectors of each patch computed earlier. This results in embedding vectors that contain both spatial information about the image regions and positional information about their locations within the image.[6], [7]

Once the embedding vectors have been created from the patches and positional information has been augmented through Positional Embedding, they are passed through a sequence of Transformer layers. Each Transformer layer consists of a Multi-head Attention layer, feed-forward neural network layers, and add-norm operations.[6], [7]

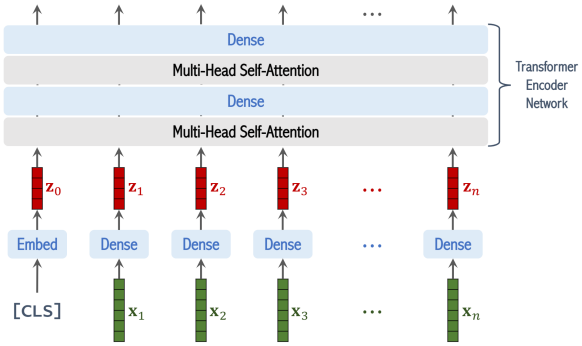

Figure 5. The self-attention layer is a key component in creating a block within the Transformer Encoder[6], [7], [8]

Self-Attention Layer

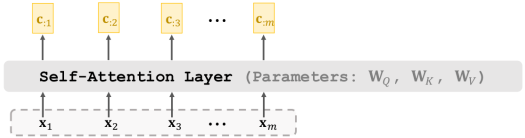

Figure 6. Self-Attention layer structure[6], [7]

Input, Output and Parameters

Input to the Self-Attention layer is a sequence $X = [x_1, x_2, x_3, ..., x_m]$.

Output of the Self-Attention layer is a context vector C containing the most important information of the input sequence $C = [c_1, c_2, c_3, ..., c_m]$.

The parameters of this layer include W_Q, W_K, W_V matrices which:
1. Query Weight Matrix (W_Q): Used to transform input vectors into query vectors. Each row of this matrix represents a query vector.
2. Key Weight Matrix (W_K): Used to transform input vectors into key vectors. Each row of this matrix represents a key vector.
3. Value Weight Matrix (W_V): Used to transform input vectors into value vectors. Each row of this matrix represents a value vector.

Operating Procedure of a Self-Attention Layer[6], [7]:

- Step 1: For each input token x_i of the input sequence X, compute the corresponding queries q_i, k_i, and values v_i using the formulas: $q_i = W_Q x_i$, $k_i = W_K x_i$, $v_i = W_V x_i$.
- Step 2: Calculate the alignment scores associated with x_i using the formula: $\alpha_i = \text{Softmax}(K^T q_i)$.
- Step 3: Compute the context vector C corresponding to x_i using the formula: $c_i = \alpha_{11} v_1 + \alpha_{21} v_2 + ... + \alpha_{m1} v_m = V\alpha_i$.

Simply put, Multi-head Attention is the stacking of multiple self-attention layers. In one Multi-head Attention layer, there are l self-attention layers. If the output dimension of each self-attention layer is $d \times m$, then the output dimension of the multi-head attention will be $(ld) \times m$. Ultimately, a Multilayer Perceptron (MLP) block takes the input context vector, denoted as c, returned from the Transformer Encoder, and yields the final outcome as probabilities corresponding to various classes.[6], [7]

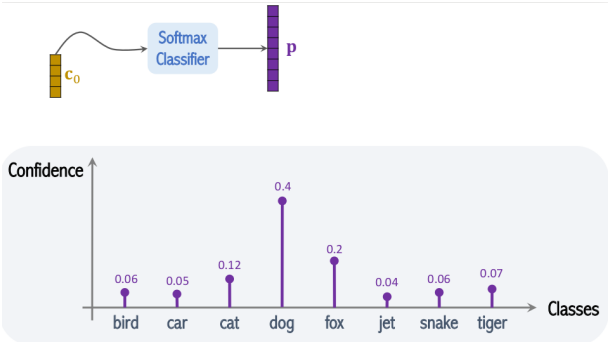

Figure 7. Softmax Classifier and Prediction[7], [8]

This architecture enables the processing of images in the form of vector embeddings, opening up the possibility of applying transformer techniques for image classification and processing.

IV. Experiments

A. Data Description

The compound data was obtained from MangoLeafBD dataset [9] and some captured by ourselves in Vietnam.

The MangoLeafBD dataset consists of 240x320 mango leaf images in JPEG format, totaling 4000 images. Among these, approximately 1800 are distinct leaf images, while the rest were prepared by zooming and rotating as necessary. Seven diseases are considered in the dataset: Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, and Sooty Mould. There are eight classes in total, including the healthy category. Each category contains 500 images. The data was collected from four mango orchards in Bangladesh: Sher-e-Bangla Agricultural University orchard, Jahangir Nagar University orchard, Udaypur Village mango orchard, and Itakhola Village mango orchard[9].

Our own captured data comprises 111 images from Southern Vietnam in 384x512 and JPG format. These images represent a single class of normal mango leaves found in Vietnam. Some images were zoomed and rotated for variation. They were distributed around locations near the residence of ours and were captured for documentation purposes.

In total, there are 4111 images, divided into training, validation, and test sets with a ratio of 70/15/15. The training set contains 2867 images, the validation set contains 610 images, and the test set contains 615 images. The training and testing were conducted on a system with the following specifications: 16GB RAM, AMD Ryzen 7 5800H with Radeon Graphics (16 CPUs) ~3.2GHz, and NVIDIA GeForce RTX 3060 laptop GPU.

B. Model and Training Parameters

We used parameters for the VIT model and parameters for training[10].

Model Parameters:

- `image_size`: This is the size to which the image input will be resized. A size like 40x40 pixels is quite small, and one can choose to work with smaller images or thumbnails.
- `patch_size`: In the Vision Transformer, the input image is divided into patches, and this parameter determines the size of each patch. A patch size of 4x4 pixels implies that each patch will be a 4x4 pixel square.
- `num_patches`: This parameter calculates the number of patches in the input image based on the patch size and the provided image size.
- `projection_dim`: This is the dimension of the projected embeddings for each patch. In Transformers, input patches are often projected into higher dimensional space to facilitate attention mechanisms.
- `num_heads`: This parameter specifies the number of attention heads in the multi-head attention mechanism. Each head attention focuses on different parts of the input chain, allowing the model to focus on different aspects simultaneously.
- `transformer_units`: This parameter determines the size of the feedforward layers in each transformer block. The first value is the size of the hidden layer, and the second value is the output size.
- `transformer_layers`: This parameter indicates the number of transformer layers stacked in the model. Each layer usually consists of multi-head self-attention blocks and feedforward neural networks.
- `mlp_head_units`: This parameter determines the size of the dense layers in the final classifier. It represents the architecture of the classifier that receives the output of the Transformer and produces the final predictions.

Training Parameters:

- `weight_decay`: This parameter is often used to control weight overshoot during optimization. It is added to the loss function to minimize overfitting. A small value such as 0.0001 is often used to apply a certain level of regularization.
- `learning_rate`: Is the rate at which the model's weights are updated during training based on the gradient of the loss function. A value of 0.001 is a common value for learning rate in many deep learning models, but may need to be adjusted based on the specific data set and architecture.
- `batch_size`: This is the number of samples that will be passed through the network during each training session. A value like 72 indicates that 72 images (or data samples) will be processed in each training iteration.
- `num_epochs`: This is the number of times the entire dataset will be forwarded and backward through the neural network during training. A value of 20 implies that the model will be trained on the entire dataset 20 times.

C. Training Process

During the training process, data is enhanced using Data Augmentation with a number of transformations such as:

- Normalization: Normalize pixel values to the range [0, 1].
- Resizing: Resize the image to the size `image_size` x `image_size`.
- RandomFlip: Flip the image horizontally randomly.
- RandomRotation: Rotate the image randomly with a certain coefficient.
- RandomZoom: Enlarge or reduce the image randomly.

We have tested many different parameters as well as added many other transformations such as RandomCrop, RandomTranslation, RandomContrast, RandomFlip("vertical"), etc. These are the parameters that produce the best results. We used the Vision Transformer (ViT) model for training. During model training, we compiled the model with settings such as optimization, loss function, and evaluation metrics. In this case, we used the `AdamW` optimization, the loss function is `SparseCategoricalCrossentropy`, and the evaluation metric is `SparseCategoricalAccuracy`. We prepared the following callbacks:

- `ModelCheckpoint`: Used to save the weights of the best model during training based on performance on the validation set.
- `EarlyStopping`: Used to stop the training process early if there is no significant improvement in minimizing the loss function on the validation set after a specified number of epochs.

Training is performed by calling the `fit` method on the model. During each epoch, the training data is divided into batches of size `batch_size`, and the model is updated with weights based on each batch. Each time after completing an epoch, the model is evaluated on

the validation set to evaluate its performance on new data, and to evaluate whether the model is learning anything from the training data and has overfitting problem. Callbacks will be called to save the best weights and check the early stopping condition. This process is repeated over each epoch until the specified number of epochs has been reached or when `EarlyStopping` is activated. When training is finished, the model is evaluated on the test set to evaluate its performance on new data it has never seen. The model is used to predict labels for each sample in the test set. These predictions are then compared with the actual labels to calculate the loss and evaluation metrics, just like in the validation process. The model is then trained and can be used to predict on new data. We have metrics that are used to evaluate model performance.

D. Evaluation Metrics

There are 5 metrics that we use to review and evaluate the model: Loss, Accuracy, Precision, Recall and F1-score.

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

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

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Loss is an important measurement to evaluate the degree of deviation between the model's predicted value and the actual value during training. During training, the goal is to minimize loss so that the model can learn the relationship between input and output.

Accuracy (1) is the ratio between the number of correct predictions and the total number of samples. It provides an overview of the model's ability to accurately classify across the entire dataset.

Precision (2) measures the proportion of true positive predictions among the total positive predictions made by the model.

Recall (3), also known as Sensitivity or True Positive Rate, measures the proportion of true positive predictions among the total actual positive cases.

F1-score (4) is the harmonic mean of precision and recall. It provides a balance between precision and recall.

Confusion Matrix provides a detailed view of the model's performance in classification each class separately. It shows the number of correct and incorrect predictions for each class, helping to determine which samples the model predicted correctly and which incorrectly.

Classification Report provides detailed information about accuracy, recall, and F1-score for each predicted class. It provides an overview of the model's performance on each class separately.

We also use color in text to show the predict result, image of experimental results with correct prediction condition in green and wrong prediction condition in red *(see subsection F)*.

E. Model Results and Comparison

The comparative analysis presented in the classification report underscores the superior performance of the Vision Transformer (ViT) model over some traditional CNN architectures. The ViT model exhibits an impressive accuracy of 98.4%, while with EfficentNet and VGG-16 (k=10), the number are respectively 75.6% and 73%, significantly outperforming the alternatives. Moreover, metrics such as precision, recall, and F1-score also reflect the superiority of ViT, indicating its robustness in making accurate predictions across various classes. Although only VN_Normal_Leaf and Healthy are absolutely correct, compared to the overall performance in other classes, it's still close to absolute correctness with the lowest rate being 0.93 for Sooty Mould. When comparing it, both VGG-16 and EfficentNet models also exhibit unstable class ratios. If applied in reality, there's a high likelihood of misprediction for various diseases. Based on the Loss curves we have presented, it shows that the training model has significantly minimized loss as well as effectively controlled overfitting.

In reality, the results may not be satisfactory if we train the model from scratch on a small dataset. This could be due to the reason that only MLP layers have local properties and translational equivariance, whereas Transformer layers are entirely globally focused.

[1]Results from abdelrahmanramadan2/Kaggle
[2]Results from rudra28121patel/Kaggle

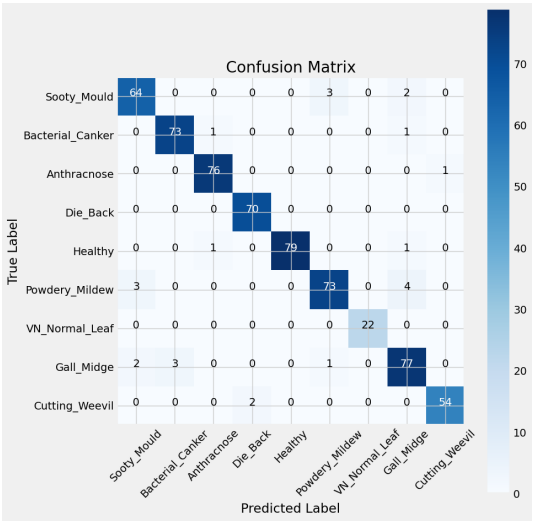

Figure 8. Confusion Matrix of the model

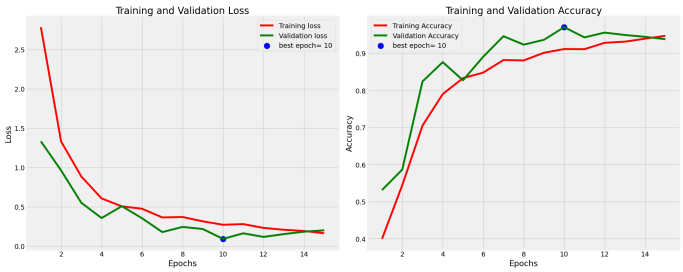

Figure 9. Loss and Accuracy

Table I
CLASSIFICATION REPORT OF THE VGG-16 (K = 10).[1]

Class	Precision	Recall	F1-Score	Support
Anthracnose	1.00	0.18	0.30	101
Bacterial Canker	0.89	0.96	0.92	90
Cutting Weevil	1.00	1.00	1.00	93
Die Back	0.81	1.00	0.89	123
Gall Midge	0.91	0.56	0.69	91
Powdery Mildew	0.88	0.91	0.89	86
Sooty Mould	1.00	0.33	0.50	112
Healthy	0.73	0.94	0.82	104
Micro Average	0.86	0.73	0.79	800
Macro Average	0.90	0.73	0.75	800
Weighted Average	0.90	0.73	0.75	800
Samples Average	0.73	0.73	0.73	800

Table II
CLASSIFICATION REPORT OF THE EFFICIENTNET MODEL[2]

Class	Precision	Recall	F1-Score	Support
Anthracnose	0.77	0.40	0.52	43
Bacterial Canker	0.82	0.76	0.79	37
Cutting Weevil	0.93	1.00	0.97	42
Die Back	0.88	0.91	0.89	46
Gall Midge	0.57	0.83	0.67	36
Powdery Mildew	0.57	0.80	0.67	30
Sooty Mould	0.78	0.49	0.60	43
Healthy	0.78	0.88	0.83	43
Accuracy			0.76	320
Micro Average	0.76	0.76	0.74	320
Weighted Average	0.77	0.76	0.75	320

Table III
CLASSIFICATION REPORT OF THE VIT MODEL

Class	Precision	Recall	F1-Score	Support
Anthracnose	0.97	0.99	0.98	77
Bacterial Canker	0.96	0.97	0.97	75
Cutting Weevil	0.98	0.96	0.97	56
Die Back	0.97	1.00	0.99	70
Gall Midge	0.91	0.93	0.92	83
Powdery Mildew	0.95	0.91	0.93	80
Sooty Mould	0.93	0.93	0.93	69
Healthy	1.00	0.98	0.99	81
VN_Normal_Leaf	1.00	1.00	1.00	22
Accuracy			**0.96**	613
Micro Average	**0.96**	**0.96**	**0.96**	613
Weighted Average	**0.96**	**0.96**	**0.96**	613

Despite ViT's remarkable performance, it is essential to acknowledge its limitations. Based on the analysis of the confusion matrix and classification report, it has come to light that the Vision Transformer (ViT) model has exhibited instances of misclassification particularly concerning diseases such as Anthracnose, Bacterial Canker, Cutting Weevil, among others. Instances of misclassification for diseases highlight the need for further refinement and fine-tuning of the model.

These misclassifications underscore the need for further refinement and augmentation of the model's training data, feature extraction methods, and potentially the architecture itself to enhance its ability to accurately discern between subtle variations and manifestations of these diseases in agricultural contexts.

F. Output Visualization

We have provided images comparing the actual results with the predictions which is positive result is right predict and negative result is wrong predict made by the model to give you the closest and most vivid understanding. These are the images from the test set that we initially randomly selected, resulting in 15 positive cases and 15 negative cases.

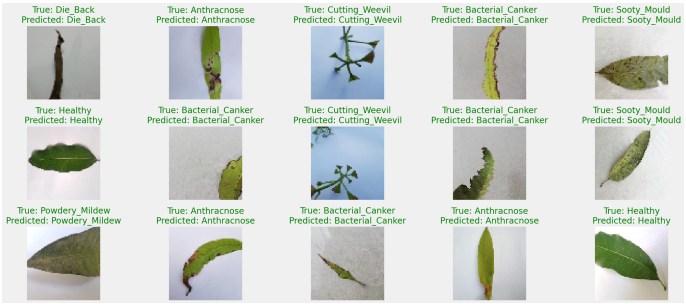

Figure 10. Positive Result

In the case of positive results, as you have observed, the model is able to make accurate predictions regardless of whether the leaves are partially captured, tilted, vertical, or whether they are young or mature. This also serves as evidence that the vision transformer model is suitable for the given problem.

Figure 11. Negative Result

However, there are some exceptional cases where the model made incorrect predictions as highlighted in figure 11. For instance, Gall Midge if captured too closely, Sooty Mould if not fully captured, Powdery Mildew if the leaf is not fully captured, etc. This suggests that the dataset we tested may contain some special cases that the model was not trained well on. Additionally, as we mentioned in the Model Results and Comparison section, MLP layers have local properties and translational equivariance.So we plan to utilize a pre-trained ViT model for fine-tuning on our data. Typically, we pre-train ViT on large datasets and fine-tune it for smaller downstream tasks. For this purpose, we remove the pre-trained prediction head and attach a zero-initialized D x K feedforward layer, where K is the number of downstream classes.[6]

V. CONCLUSION

Our article utilizes Vision Transformer technology to address the issue of diagnosing diseases in mango trees, presenting a powerful method capable of accurately identifying harmful diseases affecting the trees. This method excels due to the comprehensive capabilities of the Vision Transformer in image data recognition, providing a broader perspective compared to conventional CNN models.

The test results on our compound datasets have demonstrated the effectiveness of the approach, achieving reasonably stable accuracy. The results is astonishing with 96% accuracy, outperform other CNN models such as efficentNet and VGG-16. However, there are still limitations such as misclassifications, especially for specific diseases like Powdery Mildew, Anthracnose, Mold and healthy condition.

Future directions proposed include expanding the sample size for each class, fine-tuning hyperparameters and model architecture to optimize performance, as well as refining preprocessing techniques to minimize classification errors. Additionally, we suggest broadening the scope of research to include analyzing other factors of mango trees such as trunk and surrounding environment, as well as considering video data.

In summary, by implementing these strategies and conducting thorough validation, we hope to enhance the efficiency of mango tree disease diagnosis and achieve superior results in the future.

VI. ACKNOWLEDGMENTS

We extend our heartfelt gratitude to our esteemed colleagues from Group 4 of the AI1809/DAP391m class, whose collaborative spirit and support greatly enriched this endeavor:

- Nguyen Tran Manh Thang
- Nguyen Trong Tin
- Le Tan Phat
- Le Thanh Phong
- Nguyen Truong Thang
- Nguyen Phan Tuan Anh

Furthermore, we express our sincere appreciation to our esteemed classmates in the AI1809/DAP391m class for their unwavering support, invaluable feedback, and constructive critiques throughout our journey. We are profoundly grateful for their unwavering camaraderie and collaborative spirit, which have truly enriched our collective learning experience and propelled us towards greater heights of achievement.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[2] A. Hatamizadeh, H. Yin, G. Heinrich, J. Kautz, and P. Molchanov, "Global context vision transformers," in *International Conference on Machine Learning*. PMLR, 2023, pp. 12 633–12 646.

[3] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[4] S. He, W. Liao, H. R. Tavakoli, M. Yang, B. Rosenhahn, and N. Pugeault, "Image captioning through image transformer," 2020.

[5] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "Vt-adl: A vision transformer network for image anomaly detection and localization," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. IEEE, Jun. 2021. [Online]. Available: http://dx.doi.org/10.1109/ISIE45552.2021.9576231

[6] 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.

[7] L. M. Chien, "Vision transformer for image classification," 2021. [Online]. Available: https://viblo.asia/p/vision-transformer-for-image-classification-ORNZqV78l0n

[8] S. Wang, "Vision transformer (vit)," 2021. [Online]. Available: https://github.com/wangshusen/DeepLearning/blob/master/Slides/10_ViT.pdf

[9] S. I. Ahmed, M. Ibrahim, M. Nadim, M. M. Rahman, M. M. Shejunti, T. Jabid, and M. S. Ali, "Mangoleafbd: A comprehensive image dataset to classify diseased and healthy mango leaves," *Data in Brief*, vol. 47, p. 108941, 2023.

[10] K. Salama *et al.*, "Image classification with vision transformer," *Keras. io*, 2021.