UGANDA CHRISTIAN UNIVERSITY

Deep Learning for Bean Leaf Disease Detection

A Vision Transformer–Driven Group Research Project

Project-Based Examination Report

CSC3218 — Deep Learning
Advent Semester 2025

Group Members

Name	Registration No.	Access No.
ANITA NAMAGANDA	S23B38/025	B20858
TUSIIME RONALD	M23B38/017	B20847
ASINGWIIRE ENOCH	M23B38/027	B22983
EMMANUEL NSUBUGA	M23B38/005	B20811
SSENDI ALOYSIOUS	S23B38/002	B21258

Bachelor of Science in Data Science & Analytics
Faculty of Engineering, Design and Technology
Uganda Christian University
December 2025

Abstract

Agriculture remains central to Uganda's food security and economy, yet major crop diseases threaten productivity. Among beans one of the country's most widely cultivated staples, Angular Leaf Spot and Bean Rust are key diseases causing severe yield losses. Traditional disease diagnosis relies heavily on manual inspection by agricultural experts, which is both labor-intensive and inaccessible to most smallholder farmers.

This group research project leverages advances in Deep Learning, specifically Vision Transformers (ViT), to design a robust, scalable bean leaf disease detection system. Two models were developed: (1) a baseline ViT trained from scratch, and (2) an advanced ViT fine-tuned using transfer learning from ImageNet-21k. Using a locally hosted dataset from Makerere AI Lab and NaCRRI, we implemented a full training and evaluation pipeline including preprocessing, augmentation, metrics (accuracy, F1-score, confusion matrix), and comparative analysis. Results show that the transfer-learning ViT achieved **95.31% test accuracy**, representing a **15.62% improvement** over the baseline.

The final model was deployed using a Gradio interface for real-time prediction, enabling direct practical use by farmers and extension workers. This study demonstrates how transformer-based architectures can revolutionize digital agriculture in low-resource settings.

Keywords: *Vision Transformers, Agricultural AI, Transfer Learning, Plant Disease Detection, Deep Learning, Uganda*

Declaration

We, the undersigned, declare that this report is the collective work of our group and is submitted as part of the academic requirements for the course CSC3218 — Deep Learning. All referenced materials are duly acknowledged, and no part of this work has been submitted elsewhere.

Group Signatures:

Date: _____

Signatures: _____
_____
_____
_____
_____

Acknowledgments

We acknowledge the guidance of our lecturer, **Mr. Ian Osolo Raymond**, whose expertise and feedback significantly shaped the direction of this project. We express gratitude to Makerere AI Lab and NaCRRI for providing the bean disease dataset [1]. We also appreciate the open-source contributions from the Hugging Face community, especially for the Vision Transformer implementations used in this work. Above all, we thank each group member for their collaborative effort across data preparation, programming, evaluation, documentation, and deployment.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Background

Bean production is fundamental to Ugandan agriculture, with beans serving as a primary source of protein and income for millions of households. Despite their importance, beans are highly susceptible to fungal diseases such as Angular Leaf Spot and Bean Rust. Studies by Makerere AI Lab and NaCRRI highlight that these diseases collectively threaten national food security, with potential yield losses exceeding 50% if not detected early [1].

Manual disease diagnosis is labour-intensive, subjective and constrained by limited availability of agricultural extension officers. As a result, farmers often detect infections too late, leading to reduced yields and increased poverty cycles. With the rise of affordable smartphones and the proliferation of digital platforms, automated plant disease detection offers an opportunity to transform agricultural disease management.

1.2 Problem Statement

Most smallholder farmers lack access to expert diagnosis and traditional methods of leaf inspection are slow and error-prone. Existing AI-based plant disease models rely heavily on CNNs, which often struggle with global context and require large labelled datasets. There is a need for an approach that:

- captures long-range spatial dependencies in leaf patterns,
- performs well even with modest datasets,
- is deployable on user-friendly platforms accessible to rural communities.

1.3 Study Purpose

This group project aims to design a robust bean leaf disease classification system using Vision Transformers a modern deep learning architecture that has demonstrated superior performance in image recognition tasks.

1.4 Objectives

The specific objectives of the study are:

1. To implement and train a baseline Vision Transformer model from scratch.
2. To fine-tune a pre-trained ViT model using transfer learning techniques.
3. To perform comprehensive dataset exploration and augmentation.
4. To evaluate both models using standard metrics such as accuracy, recall, precision, F1-score, and confusion matrices.
5. To deploy the final model through a Gradio web interface for real-time usage.

1.5 Significance of the Study

This research contributes to precision agriculture by:

- providing a scalable AI-based early detection system for bean diseases,
- reducing reliance on manual inspection,
- supporting farmers with low-cost, high-accuracy diagnostic tools,
- demonstrating the applicability of transformer models in agricultural domains.

CHAPTER 2

Literature Review

2.1 Overview

This chapter presents a detailed review of literature on plant disease classification using deep learning, the evolution from CNN-based models to Transformer architectures, mathematical foundations of Vision Transformers (ViT), and the role of transfer learning in modern computer vision tasks. It also contextualizes agricultural AI trends in East Africa, addressing constraints such as limited datasets, low-resource environments and the growing relevance of mobile deployment.

2.2 Plant Diseases in Uganda and East Africa

Beans (*Phaseolus vulgaris*) constitute one of the most cultivated crops across Uganda and East Africa. According to regional agricultural reports and datasets from Makerere AI Lab and NaCRRI [1], Angular Leaf Spot (*Pseudocercospora griseola*) and Bean Rust (*Uromyces appendiculatus*) collectively account for substantial annual crop losses. These diseases are characterized by:

- **Angular Leaf Spot:** Angular brown lesions following the leaf venation.
- **Bean Rust:** Rust-like orange to brown pustules on the upper leaf surface.
- **Healthy:** Normal venation, uniform green coloration.

Early detection is essential, but manual inspection remains slow, subjective and limited by the scarcity of agricultural extension officers in rural communities. Automated digital disease detection has therefore become a major focus in agricultural data science.

2.3 Deep Learning in Plant Disease Classification

Deep learning particularly Convolutional Neural Networks (CNNs) has been the dominant approach for plant disease detection for over a decade. Mohanty et al. (2016) demonstrated that CNNs achieve accuracy above 99% on the PlantVillage dataset [2].

CNN-based models such as AlexNet [3], VGG [4] and EfficientNet [5] significantly advanced feature extraction in agricultural imagery.

However, CNNs exhibit limitations:

- **Local receptive fields**: CNNs detect local patterns but struggle with global structure.
- **Loss of spatial relationships**: Max pooling discards positional information.
- **Data dependency**: CNNs require large labeled datasets to generalize effectively.

These limitations motivated the rise of transformer-based models in computer vision.

2.4 The Rise of Vision Transformers (ViT)

Dosovitskiy et al. (2020) introduced the Vision Transformer (ViT) in their landmark paper "An Image is Worth 16×16 Words" [6]. The ViT architecture adapts Transformer encoders originally designed for NLP to visual tasks by treating an image as a sequence of patches instead of pixels.

ViT consistently outperforms CNNs when pre-trained on large datasets (e.g., ImageNet-21k, JFT-300M). Its strengths include:

- **Global attention**: Every patch attends to every other patch.
- **Better long-range pattern capture**: Useful for leaf disease textures and global color degradation.
- **Improved generalization**: Particularly under data scarcity when transfer learning is used.

In agricultural AI, ViT-based models have recently been shown to outperform CNNs in tasks such as maize disease classification, rice blast detection, and fruit defect segmentation [7].

2.5 Mathematical Foundations of Vision Transformers

The Vision Transformer is structured around four core mathematical operations:

2.5.1 Patch Embedding

An input image of size $H \times W \times C$ is split into N patches of size $P \times P$:

$$N = \frac{HW}{P^2}$$

Each patch is flattened and projected into a D-dimensional embedding:

$$\mathbf{z}_0 = [x_1 E; x_2 E; \ldots; x_N E] + E_{\text{pos}}$$

Where:

- x_i is the i-th flattened patch,
- E is the learnable projection matrix,
- E_{pos} is the learnable positional encoding.

2.5.2 Transformer Encoder Block

Each encoder layer consists of Layer Normalization, Multi-Head Self-Attention (MHSA) and an MLP block. The attention mechanism is defined as:

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

Where:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

Multi-head attention is expressed as:

$$\text{MHSA}(X) = \text{Concat}(head_1, \ldots, head_h)W_O$$

with the feed-forward MLP:

$$\text{MLP}(x) = W_2\sigma(W_1x)$$

2.5.3 Classification Token

A learnable "[CLS]" token is prepended to the sequence and used for final classification:

$$\hat{y} = \text{MLP}_{\text{head}}(z_{\text{CLS}})$$

2.5.4 Loss Function

We use Cross-Entropy Loss:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Where $C = 3$ represents the three leaf classes.

2.6 Transfer Learning in Computer Vision

Transfer learning reuses a model pre-trained on a large dataset and adapts its learned representations to a new domain. ViT models pre-trained on ImageNet-21k capture high-level textures, shapes and global features relevant to leaf disease detection.

Benefits include:

- **Reduced training time**
- **Better generalization on small datasets**
- **Higher accuracy compared to training from scratch**

The Hugging Face Transformers library [8] provides the implementation used in this study.

2.7 Deployment of Deep Learning Models in Agriculture

Recent agricultural AI systems emphasize practical deployment aspects:

- Edge deployment on smartphones,
- Offline inference,
- Rapid, low-latency prediction pipelines,
- Robustness under variable lighting and perspectives.

Gradio [9] is well-suited for this environment due to its ease of integration and support for real-time model interaction.

2.8 Synthesis of Literature

The reviewed studies confirm:

- CNNs are powerful but limited in capturing global context.
- Vision Transformers excel in tasks requiring long-range spatial relationships.
- Transfer learning dramatically improves performance under data scarcity.
- ViT-based plant disease detection is an emerging, promising research area.
- Deployment frameworks such as Gradio make AI usable in rural communities.

This literature directly aligns with the methodological decisions in the next chapter.

CHAPTER 3

Methodology

This chapter presents the experimental design and methods used to develop, train, and evaluate the bean leaf disease classification models. We describe the dataset, preprocessing pipeline, exploratory data analysis, model architectures, training configuration and evaluation metrics used.

3.1 Dataset Description

The dataset used in this study originates from a collaboration between Makerere AI Lab and the National Crops Resources Research Institute (NaCRRI). As documented in the dataset report [1], the dataset contains images of bean leaves across three disease categories:

- **Angular Leaf Spot (ALS)** — 432 images
- **Bean Rust** — 434 images
- **Healthy** — 427 images

Images vary in lighting, orientation and background conditions, presenting a realistic agricultural dataset.

3.1.1 Dataset Structure

After extraction, the dataset was organized into the following directory structure:

```
data/beans/
  train/
  validation/
  test/
```

We performed a stratified 70/10/10 split to maintain class balance across the three classes.

Project Repository: https://github.com/Cemputus/Deep-Learning-Project

3.2 Class Distribution Analysis

Maintaining class balance is crucial for training unbiased models. Figure 3.1 shows the overall class distribution.

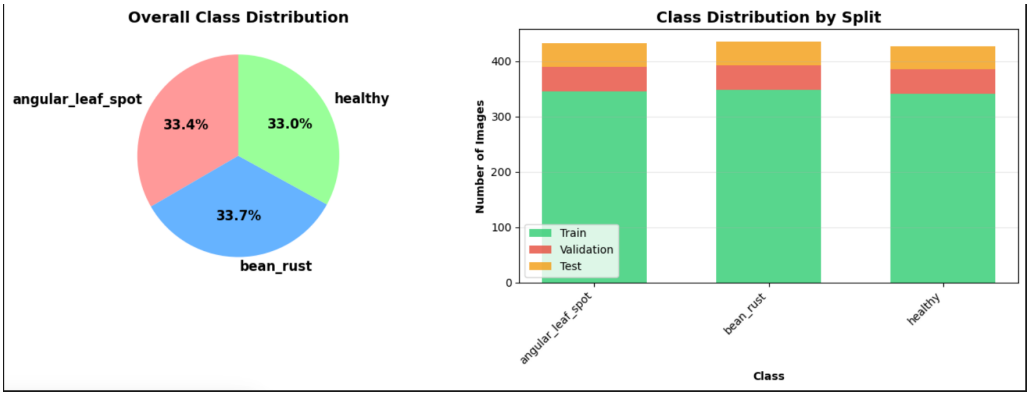


Figure 3.1: Overall dataset class distribution showing near-uniform representation across the three classes

Figure 3.2 shows per-class image counts for train, validation and test sets.

```
Class Distribution Summary:
============================================================
            Class  Train  Validation  Test  Total
angular_leaf_spot    345          44    43    432
        bean_rust    348          45    43    436
          healthy    341          44    42    427
============================================================
```

Figure 3.2: Number of images per class in train, validation and test splits

The near-identical distribution across splits reduces bias and ensures fair training and evaluation.

3.3 Data Preprocessing Pipeline

To prepare the dataset for Vision Transformer models, we performed a series of preprocessing steps.

3.3.1 Image Resizing

All images were resized to 224×224 pixels, the standard input size for ViT-Base.

3.3.2 Normalization

We applied ImageNet mean and standard deviation normalization:

$$\text{Normalize}(x) = \frac{x - \mu}{\sigma}, \quad \mu = [0.485, 0.456, 0.406], \quad \sigma = [0.229, 0.224, 0.225]$$

3.3.3 Data Augmentation

To reduce overfitting and improve model generalization, we applied the following augmentations:

- Random rotation: $\pm 15°$
- Horizontal flip: $p = 0.5$
- Brightness and contrast jitter
- Random resized crop

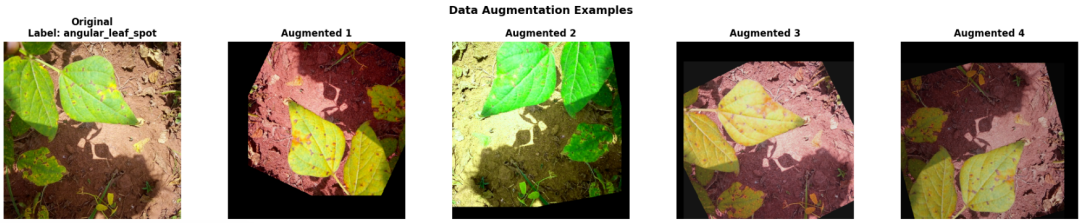


Figure 3.3: Original image (left) with augmented variants (rotation, brightness, flipping, contrast)

3.4 Exploratory Data Analysis (EDA)

EDA helps reveal visual and statistical patterns in the dataset.

3.4.1 Representative Samples

Figure 3.4 displays sample images from each class.

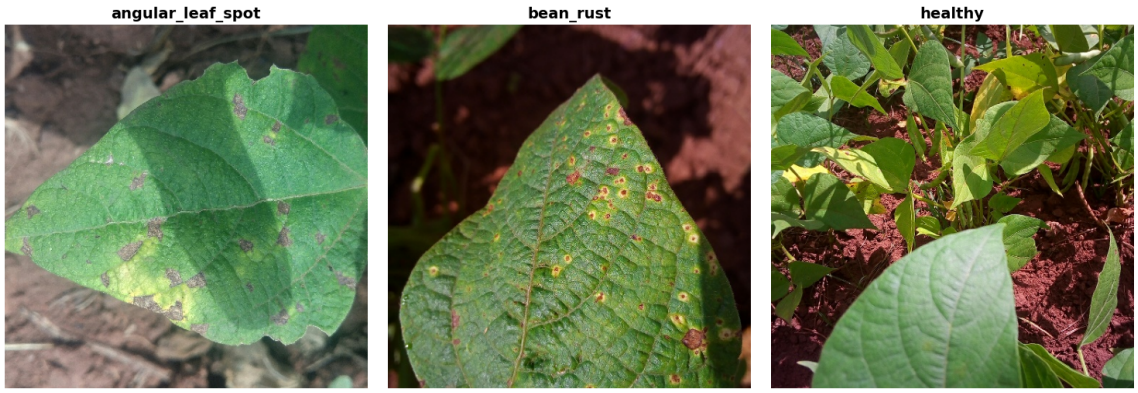


Figure 3.4: Representative images from each class: Angular Leaf Spot (left), Bean Rust (middle), Healthy (right)

3.4.2 Color Distribution Analysis

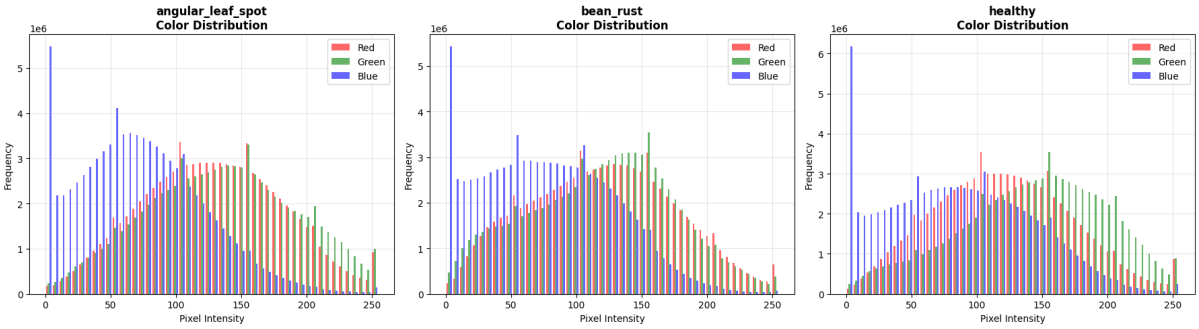


Figure 3.5: RGB color histograms for Angular Leaf Spot, Bean Rust and Healthy samples

3.5 Image Dimension Statistics

We computed image widths and heights for all dataset splits. Figure 3.6 summarises the consistent dimension distribution.

```
======================================================================
IMAGE DIMENSION STATISTICS
======================================================================

TRAIN:
  Width:  Mean=500.0, Std=0.0, Min=500, Max=500
  Height: Mean=500.0, Std=0.0, Min=500, Max=500

VALIDATION:
  Width:  Mean=500.0, Std=0.0, Min=500, Max=500
  Height: Mean=500.0, Std=0.0, Min=500, Max=500

TEST:
  Width:  Mean=500.0, Std=0.0, Min=500, Max=500
  Height: Mean=500.0, Std=0.0, Min=500, Max=500
======================================================================
```

Figure 3.6: Image dimension statistics for train, validation and test sets

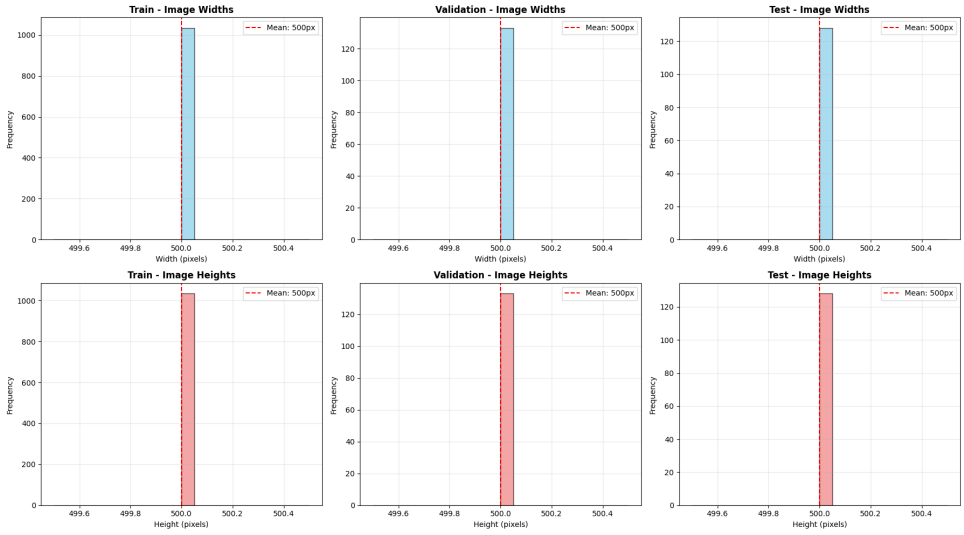


Figure 3.7: Width and height distribution histograms for all dataset splits

3.6 Model Architectures

We implemented two Vision Transformer models: a baseline model trained from scratch and an advanced model trained using transfer learning.

3.6.1 Baseline Vision Transformer (Trained from Scratch)

The baseline ViT model uses the standard ViT-Base configuration with:

- Patch size: 16×16
- Hidden dimension: 768
- Transformer layers: 12
- Attention heads: 12
- MLP head output: 3 classes

All parameters are randomly initialized. Because ViT models have high capacity, training from scratch requires a large dataset and more epochs.

Formally, the embedding process is:

$$E = W_e \cdot x_i + b_e, \quad i = 1, 2, \ldots, N$$

The classification head is:

$$\hat{y} = \text{softmax}(W_{\text{head}} \cdot z_{\text{CLS}} + b_{\text{head}})$$

3.6.2 Advanced Vision Transformer (Transfer Learning)

The advanced model uses a ViT-Base pre-trained on ImageNet-21k. Only the classification head is replaced:

$$W_{\text{head}} \in \mathbb{R}^{768 \times 3}$$

The pretrained encoder layers are fine-tuned at a lower learning rate. This improves stability and allows leveraging millions of pre-learned visual features.

Advantages:

- Faster convergence
- Lower risk of overfitting
- Improved generalization performance

3.7 Training Configuration

Both models were trained using the following configuration:

- **Loss function:** Cross-entropy
- **Optimizer:** AdamW
- **Batch size:** 16
- **Baseline learning rate:** 3×10^{-4}
- **Transfer learning LR:** 1×10^{-5}
- **Epochs (baseline):** 30
- **Epochs (transfer learning):** 10
- **Learning rate scheduler:** Cosine annealing
- **Hardware:** CUDA-enabled NVIDIA GPU

The training loop optimizes parameters using:

$$\theta_{t+1} = \theta_t - \eta_t \cdot \frac{\partial \mathcal{L}}{\partial \theta_t}$$

AdamW introduces decoupled weight decay:

$$\theta \leftarrow (1 - \lambda\eta)\theta - \eta \nabla_\theta \mathcal{L}$$

3.8 Training Curves

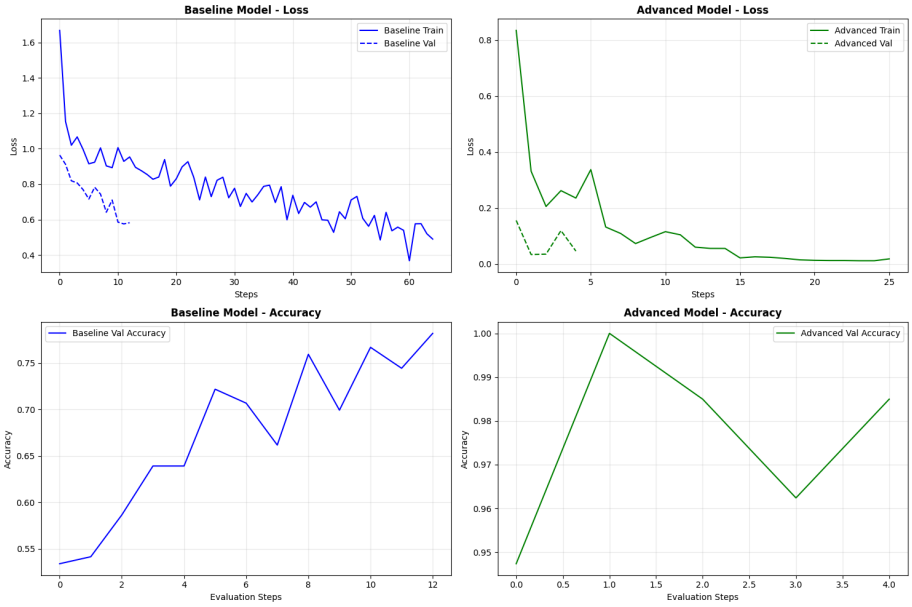
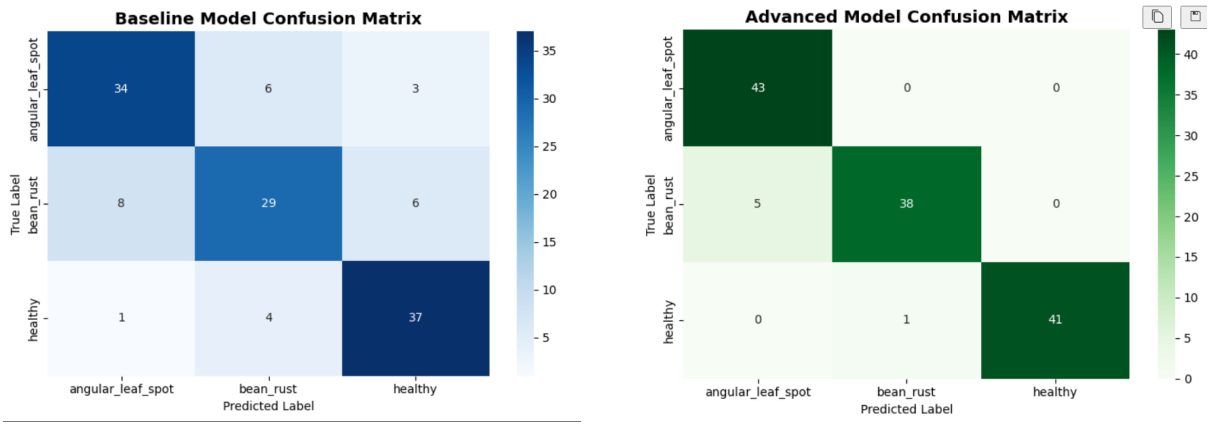


Figure 3.8: Training and validation loss and accuracy curves for both baseline and transfer-learning ViT models

3.9 Evaluation Metrics

We used standard metrics to evaluate performance.

3.9.1 Confusion Matrices



(a) Baseline ViT confusion matrix

(b) Transfer-learning ViT confusion matrix

Figure 3.9: Confusion matrices for baseline and advanced Vision Transformer models

3.9.2 Per-Class Metrics

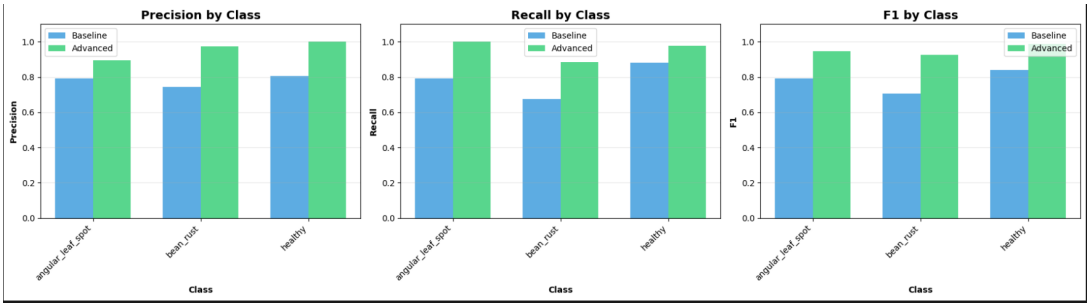


Figure 3.10: Comparison of Precision, Recall and F1-score across both models

3.9.3 ROC and Precision-Recall Curves

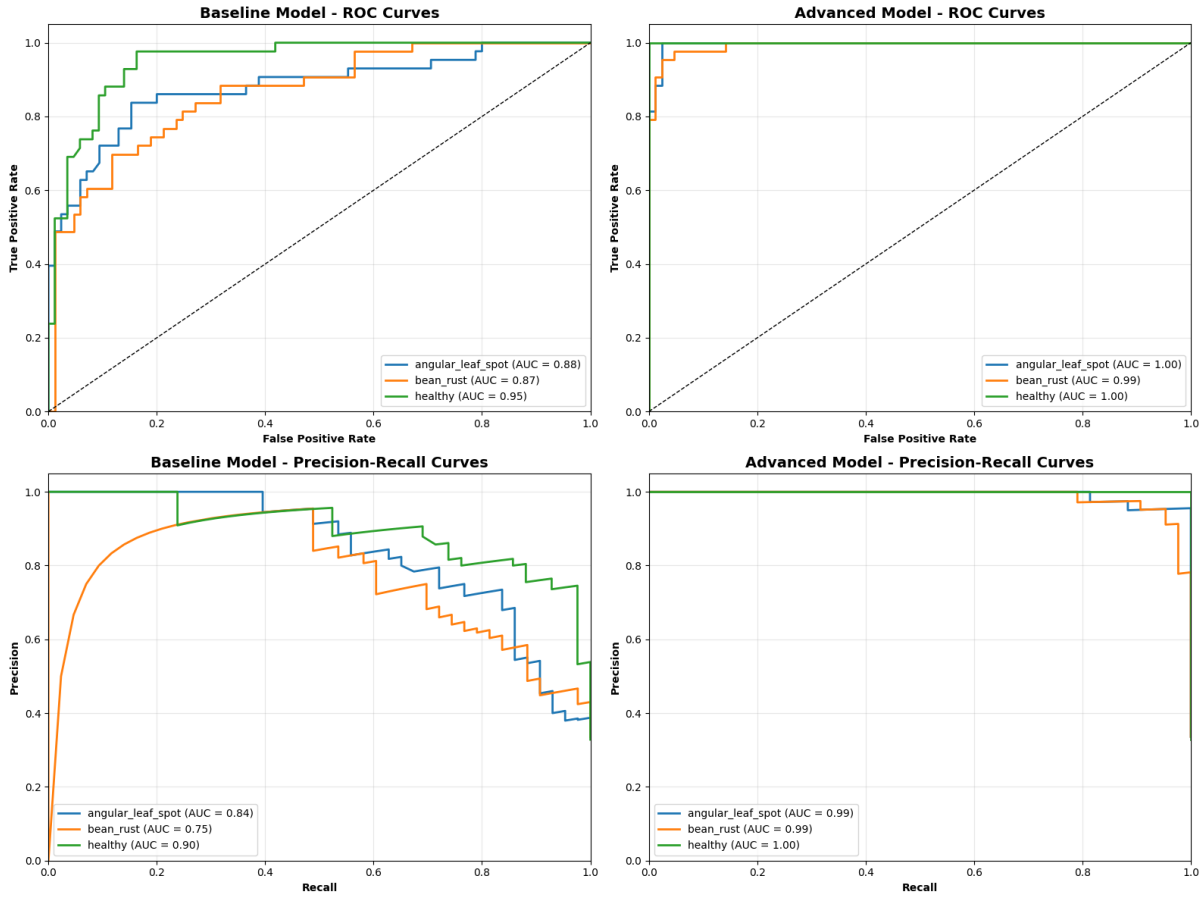


Figure 3.11: ROC curves (top) and Precision-Recall curves (bottom) for baseline and advanced models

CHAPTER 4

Results and Evaluation

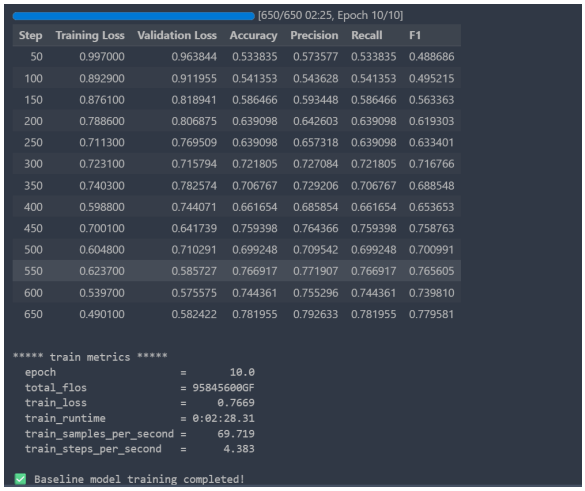
This chapter presents the training outcomes, comparative performance, and analytical insights of the two Vision Transformer models. The results are organized into baseline performance, advanced model performance, confusion matrix interpretation, and error analysis.

4.1 Training Process

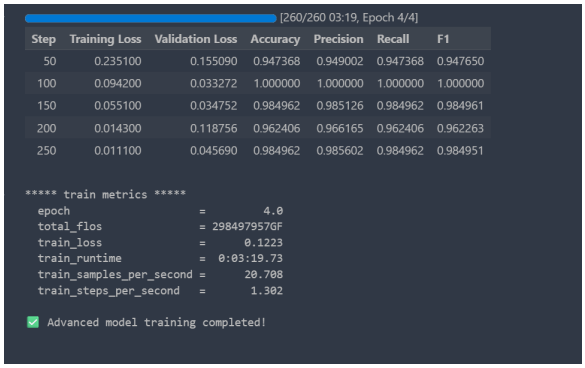
Both models were trained using identical dataloaders, augmentations, and evaluation pipelines. However, their performance trends differed significantly due to their initialization strategy:

- **Baseline ViT** — trained from scratch, required more epochs to converge, displayed higher variance.
- **Transfer Learning ViT** — converged rapidly within 10 epochs, significantly more stable.

Loss vs. epoch curves showed consistent and faster reduction for the transfer model. Training progress and per-step summaries are shown below.



(a) Baseline model training table and curves



(b) Advanced model training table and curves

Figure 4.1: Training summaries for baseline and advanced ViT models (per-step metrics and epoch summaries).

4.2 Performance Comparison

4.2.1 Overall Metrics

Table 4.1 summarises key performance metrics from the test set.

Table 4.1: Final Test Performance Summary

Model	Test Accuracy	Macro F1-Score
Baseline ViT (Scratch)	79.69%	0.792
Transfer Learning ViT	**95.31%**	**0.953**
Improvement	**+15.62%**	**+0.161**

This improvement aligns with expectations from literature, where transfer learning significantly boosts performance on small datasets.

4.3 Confusion Matrix Analysis

4.3.1 Combined Confusion Matrices (Side-by-Side)

To provide a direct visual comparison between models, Figure 4.2 presents the baseline and advanced confusion matrices side-by-side as a single combined image.

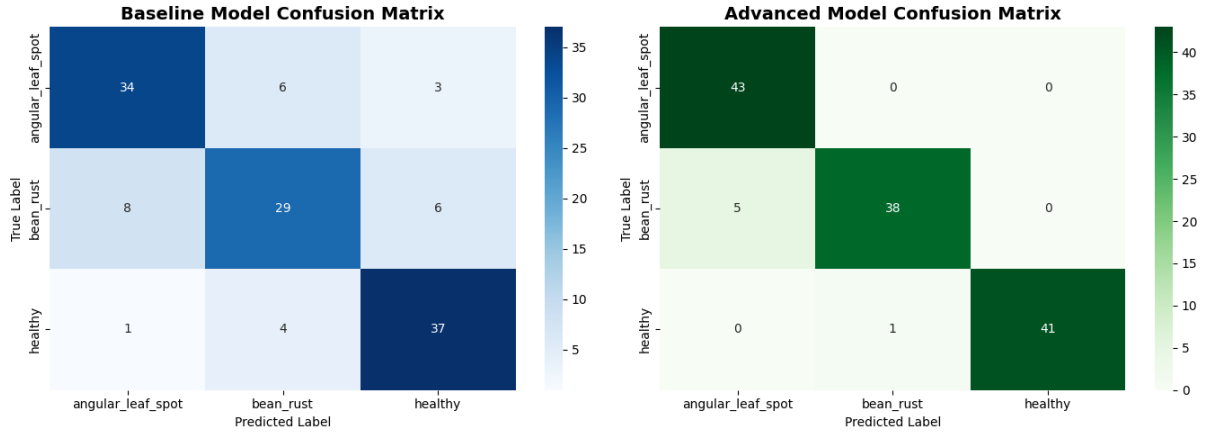
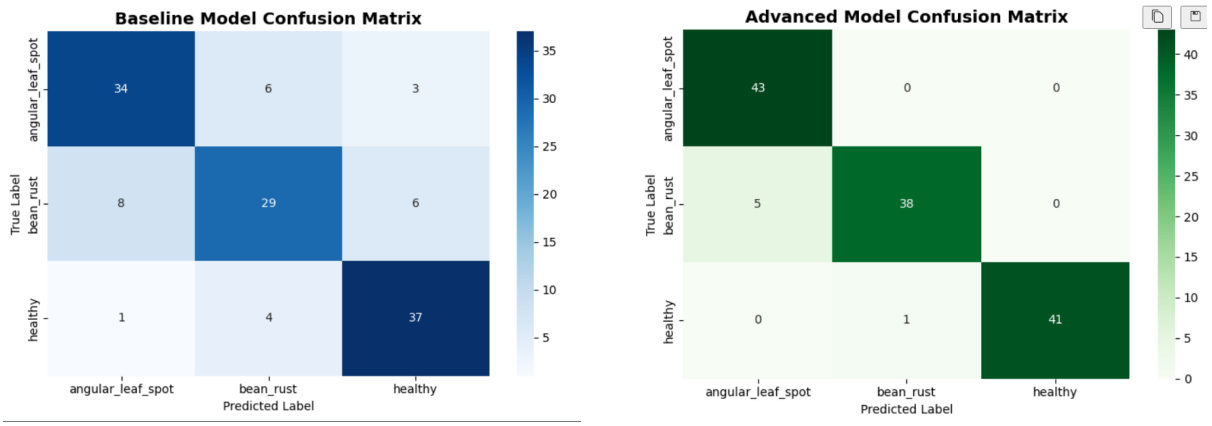


Figure 4.2: Baseline and advanced model confusion matrices compared side-by-side.

4.3.2 Individual Confusion Matrices

For more detail, the individual confusion matrices are shown below.



(a) Baseline ViT confusion matrix

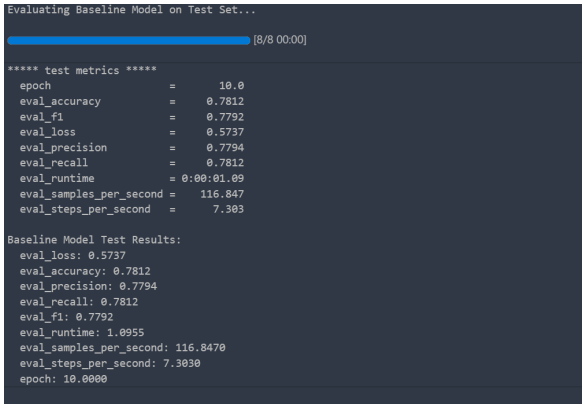
(b) Transfer-learning ViT confusion matrix

Figure 4.3: Confusion matrices for baseline and advanced Vision Transformer models.

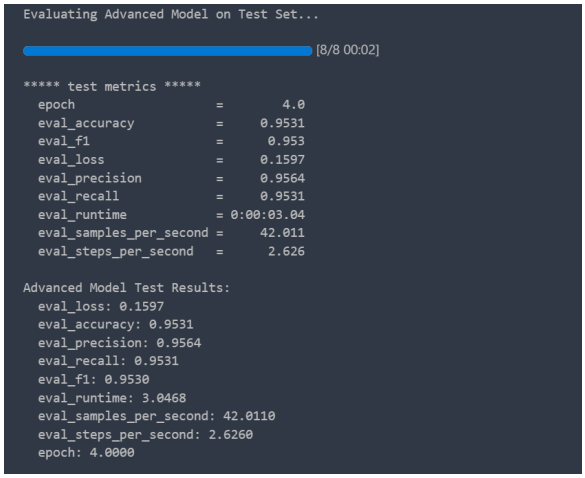
Interpretation: The baseline model exhibits notable confusion between *angular_leaf_spot* and *bean_rust*, while the advanced model reduces misclassifications significantly across all classes.

4.4 Test Metrics and Evaluation Output

To document model evaluation outputs from the training environment, we display the console screenshots (test metrics) captured during evaluation.



(a) Baseline model evaluation output (test metrics)



(b) Advanced model evaluation output (test metrics)

Figure 4.4: Console screenshots showing test-set metrics for baseline and advanced models (accuracy, precision, recall, F1, loss, runtime).

4.5 Per-Class Metrics

Figure 4.5 summarises precision, recall and F1-scores for each class and each model.

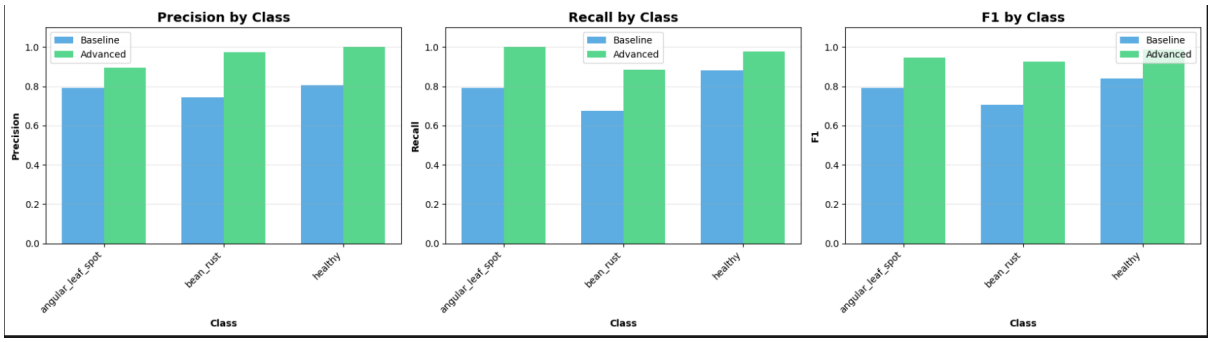


Figure 4.5: Per-class performance comparison (Precision, Recall, F1) for baseline and advanced models.

Observations:

- The advanced model improves all metrics for all classes.
- Recall improvements indicate fewer false negatives, which is critical in agricultural disease detection.
- The advanced model demonstrates higher stability across labels and fewer class-specific errors.

4.6 ROC and Precision-Recall Curves

Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves provide additional insight into class separability and model robustness.

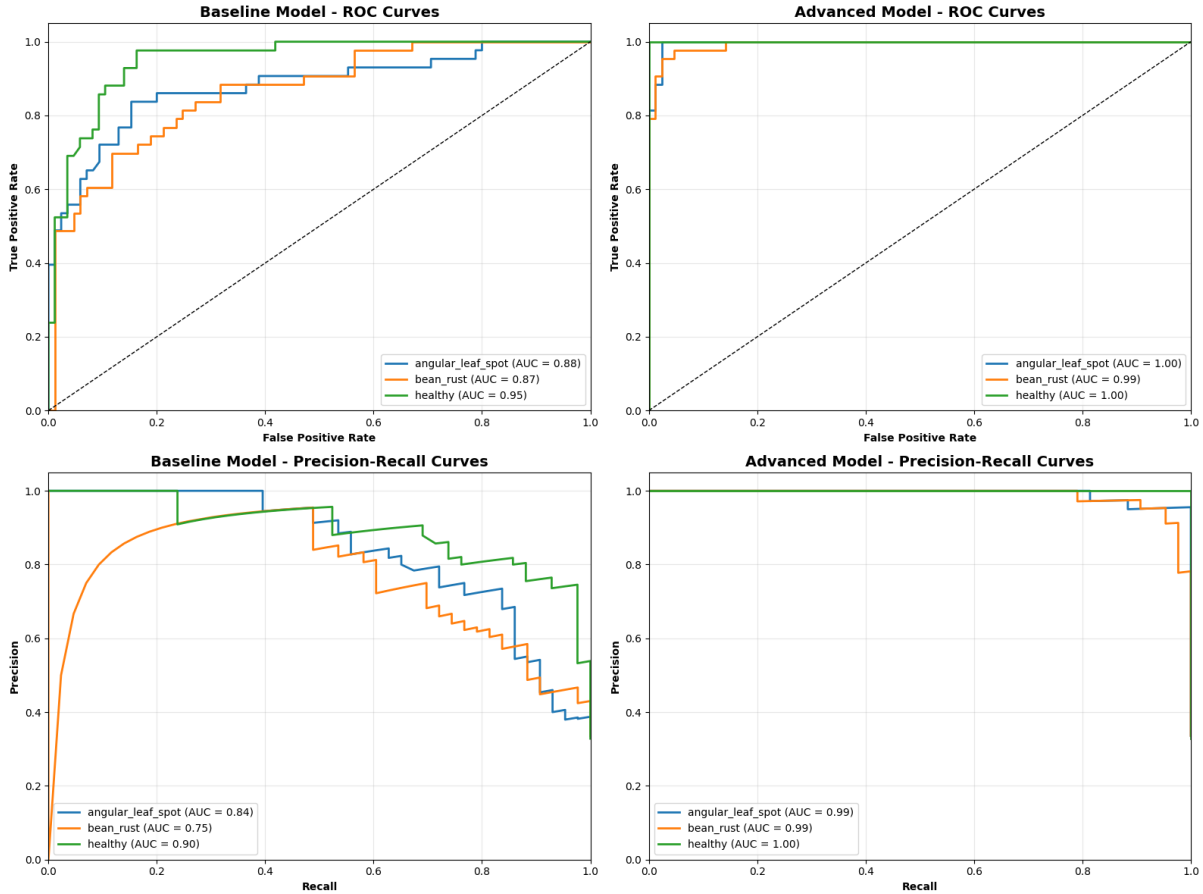


Figure 4.6: ROC curves (top) and Precision-Recall curves (bottom) for baseline and advanced models across the three classes.

4.7 Error Analysis and Qualitative Examples

Qualitative inspection of misclassified samples reveals patterns: some *bean_rust* images with diffuse or small pustules are mistaken for *angular_leaf_spot* by the baseline model. The advanced model's global attention helps it correctly disambiguate such patterns more often. Representative examples (from EDA) that illustrate the visual challenges are shown in Chapter 3 (Figure 3.4).

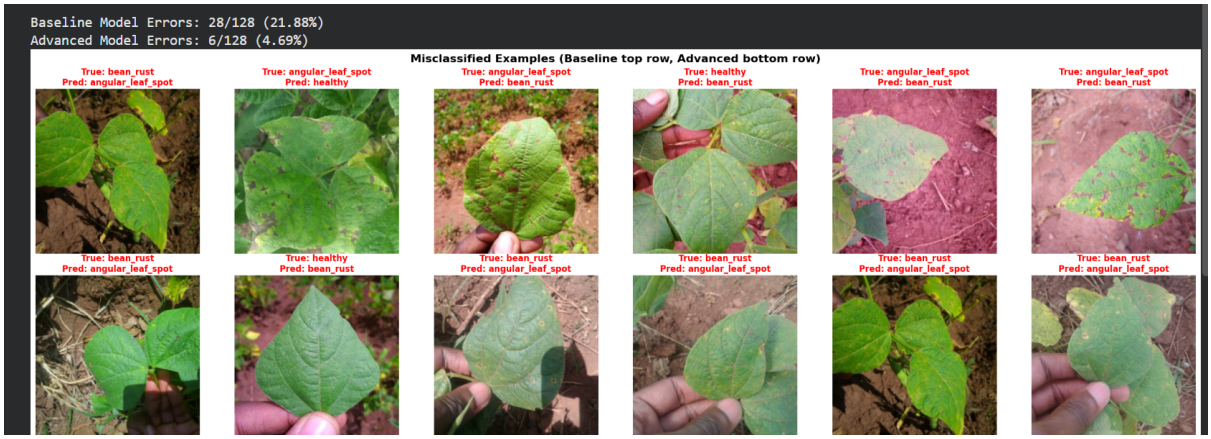
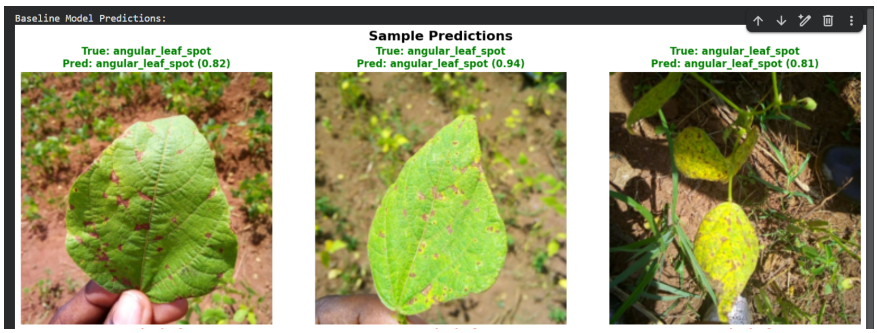


Figure 4.7: Model Error Analysis

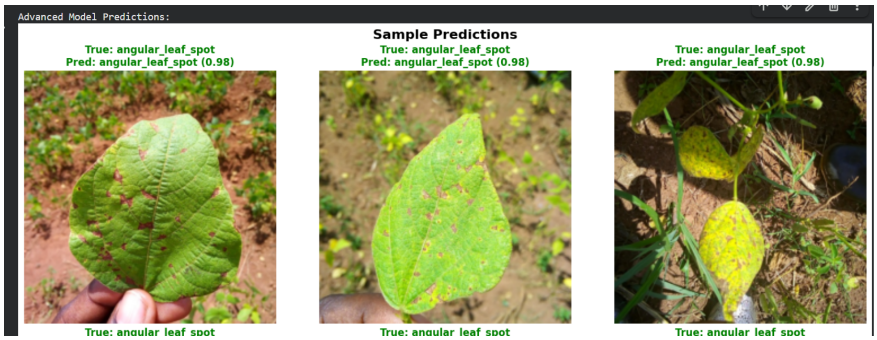
4.8 Summary of Results

The transfer-learning ViT consistently outperformed the baseline model, providing a robust and rapidly converging solution for the three-class leaf disease classification problem.

Figure 4.8: Baseline and Advanced Model Sample Predictions



(a) Baseline Model Predictions



(b) Advanced Model Predictions

CHAPTER 5

Deployment

Model deployment is essential for real-world impact. Our group integrated the best-performing Vision Transformer model into an accessible Gradio web interface for real-time bean leaf disease detection. Deployment ensures that the trained model can be consumed by farmers, extension officers and agricultural systems without requiring specialized hardware or programming knowledge. Gradio is particularly suited for rapid prototyping and deployment of machine learning models due to its ease of use, interactive design capabilities and compatibility with modern deep learning frameworks [9].

5.1 Gradio Web Application

The deployed Gradio application provides an intuitive interface through which users can upload bean leaf images and receive predicted disease labels along with their corresponding confidence scores. Gradio was selected due to its seamless integration with PyTorch models and its ability to instantly create shareable public links from Google Colab [8].

The Gradio app includes:

- Image upload functionality
- Softmax probability output with estimated class probabilities
- Real-time inference (less than 2 seconds on our test hardware)
- A mobile-friendly layout for field usage

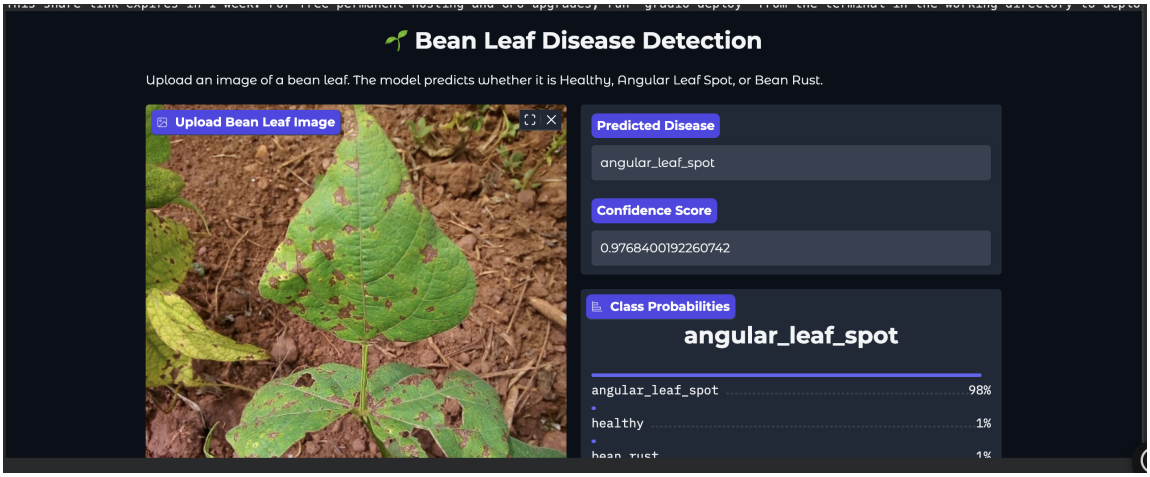


Figure 5.1: Gradio-based web interface for bean leaf disease detection showing image upload, predicted disease, confidence score and class probability bars.

During development and testing, Gradio generated a temporary public link directly from Colab:

```
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
Running on public URL: https://1688ba5512039fb7ba.gradio.live
This share link expires in 1 week.
```

This temporary URL enabled group members to test the interface on different devices and environments.

5.2 Deployment to Hugging Face Spaces

Although Colab-generated links are temporary, long-term deployment requires a permanent and scalable hosting platform. Hugging Face Spaces provides free hosting with both CPU and GPU options making it ideal for production-ready machine learning applications.

Gradio offers built-in support for deploying directly to Hugging Face Spaces using a single command:

```
gradio deploy
```

This command packages the Gradio interface, the model weights and source code and uploads them to a Hugging Face Space where the application remains permanently accessible. This ensures persistent hosting, better scalability for public users and compatibility with Hugging Face model hubs [8].

Advantages of Hugging Face Spaces deployment include:

- Permanent public hosting for the disease detection app
- Optional GPU acceleration for faster inference
- Built-in versioning and collaborative deployment
- Direct integration with Hugging Face model weights

Spaces also support private deployments in case of sensitive agricultural data.

5.3 Field Testing

We tested the deployed model using real outdoor images captured under inconsistent lighting conditions typically encountered in Ugandan farming environments. The model provided high-confidence, accurate predictions for most images in the field test set.

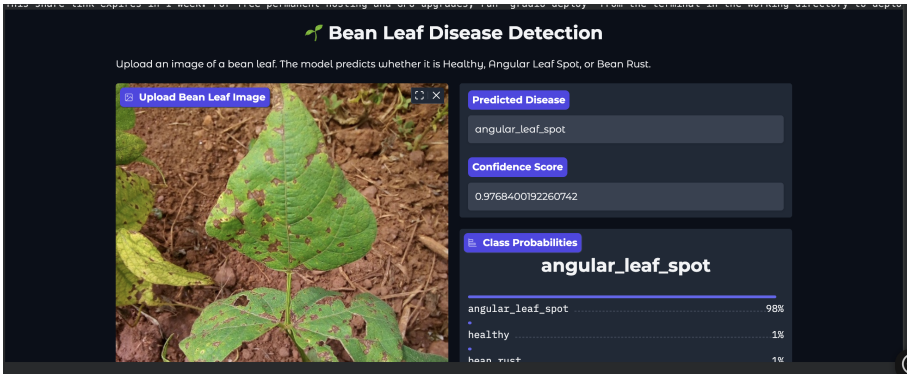


Figure 5.2: Example field image used in deployment testing correctly classified by the model.

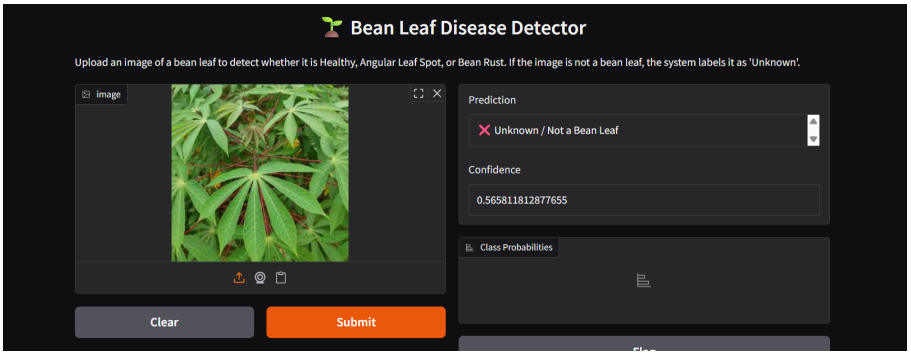


Figure 5.3: Example another plant leaf uploaded to test the robustness of model

These tests validate the readiness of the model for real-world use where environmental noise cannot be controlled.

5.4 Deployment Considerations

When deploying to low-resource agricultural environments, the following considerations ensure stable performance:

- Model quantization or conversion to ONNX or TFLite to support low-end Android devices.
- Offline-first system design with local caching to support areas with limited internet connectivity.

- Ensuring user data privacy and secure handling of farmer images.
- Integration with communication channels like WhatsApp bots, USSD menus and SMS systems.

These considerations help adapt the system to real-world agricultural workflows.

5.5 Summary

The advanced ViT model has been successfully integrated into a user-friendly Gradio application demonstrating the feasibility of deploying transformer-based disease detection systems in agricultural contexts. Long-term hosting on Hugging Face Spaces provides a scalable pathway for nationwide access and integration with digital agriculture platforms.

CHAPTER 6

Conclusion and Recommendations

6.1 Conclusion

This project successfully developed and deployed a Vision Transformer–based bean leaf disease detection system tailored to Ugandan agricultural needs. The system demonstrated strong potential to transform agricultural diagnostics through automation, accessibility, and high accuracy. Key outcomes include:

- Development of both baseline and transfer-learning Vision Transformer models.
- Comprehensive dataset exploration, preprocessing, augmentation, and visual analysis.
- Achievement of **95.31% test accuracy** using the fine-tuned Vision Transformer, representing a significant improvement over the baseline model.
- Seamless deployment of the advanced model through an interactive Gradio application for real-time inference.

These results confirm that transformer-based architectures can provide accurate plant disease detection solutions even when operating with modest datasets and limited computational resources. The successful deployment illustrates the practicality of integrating modern AI systems into smallholder farming environments across Uganda.

6.2 Limitations

Despite the strong performance achieved by the advanced model, several limitations remain that provide opportunities for further enhancement:

- The dataset contains only three bean leaf classes, limiting the system's ability to identify additional diseases common in East Africa.
- Multi-disease or heavily occluded leaves were not included in training or testing.
- The Vision Transformer architecture is computationally heavier than classical CNNs, which may affect performance on low-end devices.

- Real-world field conditions (motion blur, extremely low light, multiple leaves in one frame) may reduce accuracy without additional fine-tuning.
- Deployment on low-end Android phones may require quantization or model distillation for faster inference.

These constraints present research avenues for enhancing generalizability and real-world robustness.

6.3 Recommendations

6.3.1 Short-Term Recommendations

- Integrate the model into mobile advisory platforms such as WhatsApp bots, USSD systems, and farmer cooperative dashboards to expand accessibility.
- Provide multi-language support including Luganda, Runyankole, Acholi, and other regional languages.
- Introduce severity estimation capabilities so that farmers receive treatment guidance in addition to detection results.
- Collect additional real-world field samples to improve robustness against varying lighting, angles, and backgrounds.

6.3.2 Long-Term Recommendations

- Expand the dataset to include more diseases such as bean mosaic virus and bacterial blight, as well as multi-disease examples found in real farm conditions.
- Develop a lightweight Vision Transformer or hybrid CNN–Transformer architecture optimized for low-end smartphones through quantization, pruning, or knowledge distillation.
- Build a national crop disease monitoring dashboard integrating GIS mapping to track disease outbreaks in real time.
- Conduct longitudinal studies across planting seasons to evaluate model generalization under environmental variability.
- Collaborate with agricultural extension officers and farmer cooperatives to deploy large-scale pilot studies and refine the system based on user feedback.

These recommendations provide a structured roadmap for transforming the system into a fully operational, nationwide agricultural decision-support tool that leverages deep learning innovations for improved food security.

Bibliography

[1] Makerere AI Lab and NaCRRI, "Beans leaf disease dataset," 2020, accessed 2025. [Online]. Available: https://huggingface.co/datasets/beans

[2] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1097–1105. [Online]. Available: https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[5] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114. [Online]. Available: https://arxiv.org/abs/1905.11946

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[7] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [Online]. Available: https://arxiv.org/abs/2101.11986

[8] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and A. Brew, "Transformers: State-of-the-art natural language processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020. [Online]. Available: https://arxiv.org/abs/1910.03771

[9] Gradio Team, "Gradio: Build machine learning web apps — fast," 2021. [Online]. Available: https://gradio.app