Explainable Deep Learning Framework for Soybean Leaf Disease Detection from UAV Imagery Using Custom CNN and Pre Trained Models

Suraiya Chowdhury
ID: 2021-3-60-295
2021-3-60-295@std.ewubd.edu

Sabera Zannat Sheba
ID: 2022-2-60-095
2022-2-60-095@std.ewubd.edu

Fahimun Islam Lamiha
ID: 2021-3-60-181
2021-3-60-181@std.ewubd.edu

Sulatan Mahmud Shuvo
ID: 2021-3-60-207
2021-3-60-207@std.ewubd.edu

May 22, 2025

Abstract

This study addresses the critical challenge of early soybean disease detection using UAV-collected imagery. We developed a comprehensive framework combining deep learning classification with explainable AI to identify four conditions: healthy soyabean, soyabean semilooper pest attack, soyabean mosaic, and rust. A custom convolutional neural network CNN was developed from scratch also used several well-known pre-trained models including **ResNet-50, VGG16, MobileNetV2, EfficientNet-B3, DenseNet121, ResNet152, and DenseNet201** were applied on dataset. The training process included advanced data augmentation strategies, early stopping to prevent overfitting, and mixed-precision training (AMP) for improved computational efficiency. Model performance was evaluated with precision, recall, F1-score, and accuracy metrics on the test set. The experiments showed that transfer learning models performed the best, with **EfficientNet-B3** achieving the highest accuracy **98.60%**, followed closely by **DenseNet121-97.54%** and **DenseNet201-97.19%** and **MobileNetV2-97.19%**. The integration of explainability techniques like: **Grad-CAM, Grad-CAM++, Eigen-CAM, and LIME** revealed that the models correctly focused on disease-specific leaf features, such as color changes and spots when making predictions. The results show the effectiveness of transfer learning and XAI integration in developing robust, accurate, and transparent plant disease diagnosis systems leveraging UAV imagery.

In conclusion, the results demonstrate the effectiveness of transfer learning and XAI integration in developing robust, accurate, and transparent plant disease diagnosis systems leveraging UAV imagery.

1 Introduction

Soybean cultivation represents a cornerstone of global agriculture, providing essential protein and oil for both human consumption and animal feed. However, the productivity and sustainability of soybean crops face significant threats from various diseases and pests, which can dramatically reduce yields and quality if not detected and managed early. Manual disease diagnosis is often labor-intensive, time-consuming, and prone to subjective errors, which limits scalability, especially for large-scale crop monitoring. Deep learning methods, particularly convolutional neural networks (CNNs), have demonstrated superior performance in automating disease detection from plant leaf images, enabling rapid, accurate, and cost-effective diagnostics. Wu et al.,[1] represents that traditional monitoring approaches rely heavily on manual inspection by agricultural experts, which is labor-intensive, time-consuming, and often impractical for large-scale farming operations.

Specifically, this research addresses the classification of soybean leaf diseases using images acquired via unmanned aerial vehicles (UAVs). Soybean leaf diseases such as leaf spot and rust, pose substantial threats to agricultural productivity. Yu et al.,[2]present that the challenge is exacerbated by environmental variability, complex field backgrounds, and disease symptoms with high visual similarity, which make robust and reliable classification difficult. Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification tasks, including the identification of plant diseases from leaf images. Moreover, the interpretability of deep learning models remains limited, hindering the adoption of these methods in practical agricultural decision-making.

Despite these advances, several challenges remain in developing reliable disease detection systems for real-world agricultural applications. First, the development of custom CNN architectures optimized for specific disease detection tasks requires significant expertise and computational resources. Second, while transfer learning approaches using pre-trained models offer potential shortcuts, their comparative performance for specific agricultural applications remains understudied. Third, and perhaps most critically, most deep learning models operate as "black boxes," making their decisions difficult to interpret and trust by agricultural practitioners. Rodriguez et al.,[3] denotes that we utilize extensive data augmentation and early stopping to improve model generalization and prevent overfitting on complex real-world datasets with diverse environmental conditions.

This research addresses these challenges by developing an explainable deep learning framework for soybean leaf disease detection from UAV imagery. Specifically, we focus on accurately classifying four conditions: healthy soyabean, semilooper pest attack, soyabean mosaic, and rust. Our objectives are threefold: First, to implement a custom CNN architecture specifically designed for soybean disease classification; Second, to fine-tune and evaluate seven pre-trained models including: ResNet50, VGG16, MobileNetV2, EfficientNetB3, DenseNet121, ResNet152, and DenseNet201 using transfer learning; and then to apply multiple explainable AI (XAI) techniques including Grad-CAM, Grad-CAM++, Eigen-CAM, and LIME to provide interpretable visualizations of model decisions.

The key contributions of this work include:Firstly, a comprehensive comparison of custom and pre-trained CNN architectures for soybean disease detection, providing valuable insights for model selection in agricultural applications. Secondly, Implementation and evaluation of multiple XAI techniques that enhance the interpretability of model predictions, making the system more transparent and trustworthy for end-users. Thirdly,A robust framework that achieves high classification accuracy (up to 98.6%) while simultaneously providing visual explanations of the disease-specific features used in decision-making. Again, empirical evidence demonstrating the effectiveness of UAV-based imagery combined with deep learning for early detection of soybean diseases, potentially enabling more timely and targeted interventions.

This paper proceeds as follows: Section 2 reviews related work on deep learning and object detection approaches in plant disease detection. Section 3 presents the dataset and 4 deals with methodology. Section 5 Grad Cam , section 6 reveals Lime, section discusses experimental results, and Section 8 shows the reasults, section 9 revies the Discussion, section 10 explores the conclusion, and section 11 concludes with future research directions.

2 Related Work

Goshika et al., [4] developed a YOLOv5s deep learning object detection model trained on a dataset of 2930 near-field soybean leaf images captured with an iPhone 13 Pro. Their model classifies damage severity into five distinct levels and achieved a mean average precision (mAP) of 92%. However, their approach struggled with complex field conditions, such as overlapping leaves and variable backgrounds, which limit real-world applicability. Future work suggests enhancing robustness to such environmental complexities. In contrast, our study utilizes UAV-based soybean images combined with a custom CNN and transfer learning models enhanced with data augmentation and advanced interpretability techniques (Grad-CAM, LIME). This improves robustness to complex backgrounds, overlapping leaves, and field variability, enabling more reliable damage classification in real agricultural settings.

Slimani et al., provided a comprehensive review of drone-based plant disease detection techniques leveraging convolutional neural networks (CNNs) [5]. They highlighted models achieving up to 99.63% average precision on the widely-used PlantVillage dataset. Nevertheless, the study noted significant challenges with drone image variability, including differences in viewing angles and lighting conditions, which affect model performance in real agricultural environments. The authors emphasized the need for more robust drone image analysis methods capable of handling these real-world variabilities. Our approach explicitly incorporates comprehensive data augmentation simulating various lighting and angle conditions in UAV imagery, combined with transfer learning to improve model generalization. This enhances the model's ability to handle real-world drone image variability more effectively.

Bhargava et al.,[6] surveyed various machine learning and deep learning techniques for plant leaf disease detection. Their analysis found that models generally perform well on laboratory datasets but suffer accuracy degradation on real-field images due to complex backgrounds and uncontrolled environments. Moreover, many existing models lack interpretability, which hampers expert trust and practical deployment. They recommend future research focus on improving model robustness and explainability to bridge the gap between controlled experiments and field conditions. Our work addresses this by integrating state-of-the-art explainable AI methods (Grad-CAM, Grad-CAM++, Eigen-CAM, LIME) alongside real-field UAV datasets, enabling transparent decision-making and enhancing expert trust while maintaining strong accuracy on complex backgrounds.

Jahin et al.,[7] proposed an interpretable hybrid model combining MobileNetV2 CNN and GraphSAGE graph neural networks for soybean disease classification. Their sequential CNN-GNN architecture achieved 97.16% accuracy on a dataset containing ten soybean leaf disease classes. This model uniquely integrates spatial feature extraction with relational dependency modeling and uses Grad-CAM and Eigen-CAM for visual explanation. However, scalability to larger datasets and graph construction efficiency remain open challenges, suggesting directions for further research. While effective, our methodology focuses on a scalable, efficient custom CNN and transfer learning pipeline tailored for damage severity classification on a larger UAV dataset. This avoids potential graph-based scalability issues and emphasizes practical real-world deployment.

Al Sahili and Awad., [8] introduced AgriNet, a large-scale agricultural image dataset containing over 160,000 images across 423 classes of plant species, diseases, pests, and weeds. They pretrained five CNN architectures (VGG16, VGG19, Inception-v3, InceptionResNet-v2, and Xception) on this dataset, achieving up to 94% accuracy with AgriNet-VGG19. Despite strong performance, class imbalance and rare categories continue to challenge model robustness. Future efforts should address these imbalances and expand domain-specific pretrained models for better generalization. Our study addresses class imbalance through extensive data augmentation, and trains both custom CNN and domain-adapted transfer learning models on a focused UAV soybean dataset. We combine this with interpretability methods to enhance robustness and ensure practical applicability in the soybean domain.

Table 1: Comparison of Related Work Models, Performance, Limitations, and Our Improvements

Paper No	Model & Dataset	Accuracy	Limitations & Future Work	How this Paper Addresses these Gaps
[4]	YOLOv5s on 2930 near-field soybean leaf images	92% mAP	Struggles with overlapping leaves, complex backgrounds	Uses UAV images, custom CNN + transfer learning + XAI; more robust to complex backgrounds and overlaps
[5]	CNNs reviewed on PlantVillage + drone datasets	Up to 99.63%	Drone image variability (angles, lighting)	Data augmentation simulating angles/lighting + transfer learning for better generalization
[6]	Various ML/DL on lab and field data (survey)	Good on lab; drops on field	Accuracy drop in complex backgrounds; lack interpretability	Integrates explainable AI + real UAV data to boost trust and accuracy
[7]	Hybrid MobileNetV2 + GraphSAGE CNN-GNN (10 classes)	97.16%	Scalability and graph efficiency	Focus on scalable custom CNN + transfer learning on larger UAV dataset
[8]	AgriNet pretrained CNNs on 160k images (423 classes)	Up to 94%	Class imbalance, rare classes	Extensive augmentation + domain-adapted transfer learning + interpretability for robustness

3 Dataset and Preprocessing

3.1 Dataset Description

For this study, we utilized the Soybean UAV-Based Image Dataset,(shown in Figure1), part of the MH-SoyaHealthVision dataset. This comprehensive collection of aerial imagery was captured using unmanned aerial vehicles (UAVs) over soybean fields in the Maharashtra region of India. The dataset was created by Sayali Shinde and Dr. Vahida Attar from COEP Technological University, Pune, and IDEASTechnology Innovation Hub, Indian Statistical Institute Kolkata. The dataset contains high-resolution UAV images across four distinct

Figure 1: ample images from the soybean UAV dataset showing four disease classes: Healthy Soybean, Soybean Semilooper Pest Attack, Soybean Mosaic, and Rust

classes that iis not well-balanced as detailed in Table 2. MH-SoyaHealthVision: An Indian UAV and Leaf Image Dataset for Integrated Crop Health Assessment. DOI: 10.17632/hk-bgh5s3b7.1

For our experimental setup, we split the dataset of 2842 soybean leaf images into **training:70%, validation:20%, and testing:10%** sets — resulting in approximately 1990 images for training, 568 for validation, and 284 for testing. This allocation ensures ample data for learning, effective tuning via validation, and unbiased evaluation through the test set. All images were resized to **224×224** and normalized (mean = 0.5, std = 0.5). To improve generalization, the training set was augmented using random resized cropping and horizontal flipping.

Table 2: Dataset Distribution by Class

Class	**Number of Images**	**Percentage (%)**
Healthy_Soyabean	280	9.85%
Rust	1000	35.18%
Soyabean Semilooper_Pest_Attack	790	27.80%
Soyabean_Mosaic	772	27.16%
Total	**2842**	**100%**

3.2 Data Preprocessing and Augmentation

Before starting the model training, all images were prepared using some basic steps with Pytorch's library like torchvision.transforms. These crucial steps helped make sure the images were the same size, helped the model learn faster, and improved its ability to work well on new data.

First, every image was resized to **224 x 224** pixels, which is a common size used by many deep learning models. Then, the images were normalized by adjusting their colors so the values are centered around zero, using a **mean and standard deviation of 0.5** for each color channel. This helps the model train more smoothly. We applied data augmentation only to the training images to make the model stronger. In our procedure, we implemented **RandomResizedCrop** which randomly cuts out a part of the image and resizes it back to **224×224**, so the model learns to recognize leaves at different sizes and positions. Additionally, we used **RandomHorizontalFlip** which flips the image left-to-right randomly, helping the model learn to identify leaves even if they appear mirrored.

These changes help the model see many different versions of the leaves, making it better at handling real-world variations. For the validation and test images, we only resized and normalized them, no augmentation was used to keep evaluation fair and consistent.All of these steps used standard **PyTorch** tools without any special custom coding.

4 Methodology

The research methodology follows a systematic pipeline starting with dataset preprocessing through data augmentation and train-validation-test splitting, followed by implementation of both custom CNN architectures including with AMP and seven transfer learning models. The training process incorporates early stopping mechanisms and comprehensive evaluation through loss curves and performance metrics. Finally, explainable AI techniques including Grad-CAM, Grad-CAM++, Eigen-CAM, and LIME are applied to provide interpretable insights into model decision-making processes shown in figure-2.

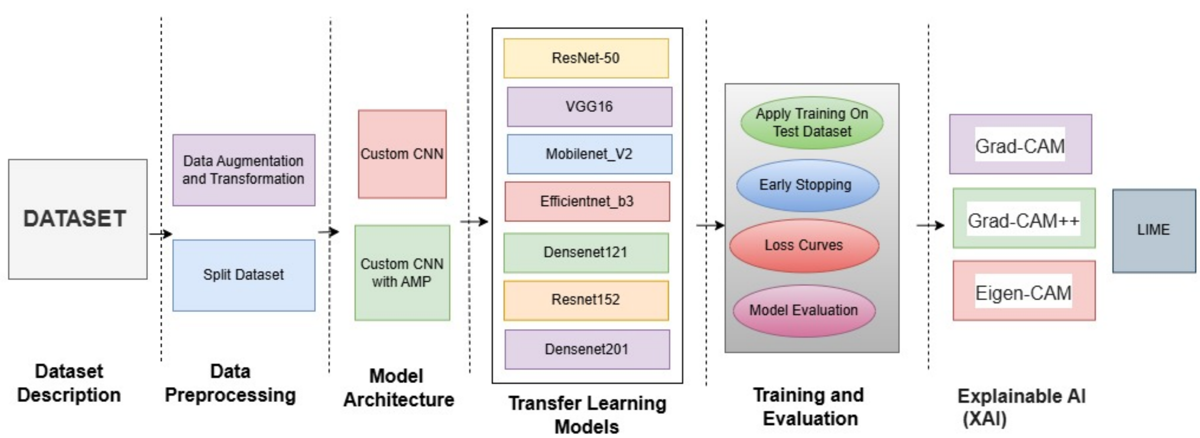

Figure 2: Flowchart of the project process

4.1 Custom CNN Architecture

Custom CNN Architecture: Our custom Convolutional Neural Network (CNN) was designed specifically for accurate disease and pest detection in soybean UAV imagery. The architecture balances depth and complexity to effectively learn features from the dataset while avoiding overfitting. Network Structure: Here six convolutional layers are activated for pooling operation.

Table 3: Detailed Layer Description of Custom CNN

Layer Type	**Filters**	**Kernel**	**Activation**	**Pooling**
Conv1	32	3×3	ReLU	MaxPooling 2×2
Conv2	64	3×3	ReLU	MaxPooling 2×2
Conv3	128	3×3	ReLU	MaxPooling 2×2
Conv4	256	3×3	ReLU	MaxPooling 2×2
Conv5	512	3×3	ReLU	MaxPooling 2×2
Conv6	1024	3×3	ReLU	MaxPooling 2×2
FC1	512	–	ReLU	–
FC2	256	–	ReLU	–
FC3	128	–	ReLU	–
FC4	64	–	ReLU	–
FC5	32	–	ReLU	–
Output	4	–	Softmax	–

Architectural Choices: Our architectural decisions were carefully considered to optimize the model's performance. Starting with **32 filters** and doubling with each subsequent convolutional layer up to **1024** allows the network to progressively learn more complex and abstract features. The increased depth captures intricate patterns such as pest damage and disease spots, essential for fine-grained classification. A consistent **3×3 kernel size** was used throughout, as it balances capturing local spatial information with computational efficiency, a proven choice in many successful CNN architectures. **MaxPooling** with a **2×2** window follows each convolutional layer to reduce spatial dimensions gradually, lowering computational load while retaining salient features.

Regularization Techniques: To mitigate overfitting given the dataset size and network complexity, we incorporated the following regularization methods. A dropout rate of 0.01 was applied after each fully connected layer. This technique randomly deactivated neurons during training, preventing co-adaptation and encouraging the network to learn more robust, generalized representations. Although not explicitly used in this version, batch normalization is recommended to stabilize and accelerate training by normalizing layer inputs. Future versions may include batch normalization layers after convolutions to further enhance model performance.

4.2 Transfer Learning Models

Seven pre-trained CNN architectures, all pretrained on the ImageNet dataset, were adapted and fine-tuned for soybean disease and pest classification from UAV imagery. The models were modified by replacing the final classification layer and selectively fine-tuning deeper layers.

ResNet50: The ResNet50 architecture, a powerful 50-layer Residual Network pretrained on ImageNet, was adapted for our soybean disease classification task. Initially, we maintained frozen convolutional layers to preserve learned features while replacing and training only the final fully connected layer. As training progressed, we strategically unfroze deeper blocks to allow the model to fine-tune its feature extraction for soybean-specific patterns. The final fully connected layer was specifically adjusted to output four classes: healthy soyabean, soyabean semilooper pest attack, soyabean mosaic, and rust. For optimal performance, we utilized a learning rate of 0.001 with Adam optimizer, training for up to 50 epochs with early stopping to prevent overfitting. The model performed exceptionally well on our UAV-based soybean dataset, effectively learning to distinguish subtle variations in leaf coloration and spotting patterns characteristic of different disease conditions.

VGG16: The VGG16 architecture, a classic 16-layer deep convolutional network pretrained on ImageNet, was configured for our soybean disease classification task. We implemented a strategic approach by initially keeping the convolutional base frozen while replacing and training only the classifier layers. The final fully connected layer was modified to output our four target classes. For training, we maintained a learning rate of 0.001 with Adam optimizer, implementing a maximum of 50 epochs with early stopping to prevent overfitting and ensure computational efficiency as the model learned to differentiate between healthy and diseased soybean patterns in UAV imagery.

EfficientNet-B3: The EfficientNet-B3 architecture, a compound-scaled CNN pretrained on ImageNet, was implemented for our soybean disease classification task. This model employs a unique compound scaling method that uniformly scales network width, depth, and resolution, resulting in balanced network growth and superior performance. We initially kept the base layers frozen while replacing only the classifier head to output our four target classes. During training, we applied selective unfreezing for fine-tuning to adapt the model to the specific characteristics of UAV-based soybean imagery. The training process utilized a learning rate of 0.001 with Adam optimizer and implemented a maximum of 50 epochs with early stopping to prevent overfitting while the model learned to distinguish healthy soybean leaves from various disease conditions.

MobileNetV2: The MobileNetV2 architecture, a lightweight CNN pretrained on ImageNet, was adapted for our soybean disease classification system. This model is notable for its inverted residual structure and linear bottlenecks, making it computationally efficient while maintaining high accuracy. We initially froze the convolutional base while replacing and training only the classifier component. As training progressed, we selectively fine-tuned deeper layers to optimize feature extraction for soybean leaf characteristics. The final layer was modified to output our four target classes. For training, we employed a learning rate of 0.001 with Adam optimizer and implemented a maximum of 50 epochs with early stopping to ensure efficient convergence while the model learned to identify different soybean conditions in UAV imagery.

DenseNet121: The DenseNet121 architecture, pretrained on ImageNet and characterized by its densely connected convolutional layers, was implemented for our soybean disease classification task. This network features direct connections from each layer to all subsequent layers, strengthening feature propagation and encouraging feature reuse. We initially kept the base layers frozen while replacing and training only the classifier component. During training, we applied partial fine-tuning to optimize the model for soybean disease identification. The final classification layer was modified to output our four target classes. Training was conducted using a learning rate of 0.001 with Adam optimizer and implemented a maximum of 50 epochs with early stopping to prevent overfitting while the model learned to differentiate between healthy and diseased soybean patterns.

ResNet152: The ResNet152 architecture, a deep residual network with 152 layers pretrained on ImageNet, was adapted for our soybean disease classification task. This ultra-deep network uses residual learning to overcome the degradation problem, allowing effective training of substantially deeper networks. We employed a similar freezing strategy as with ResNet50, initially keeping the convolutional layers frozen while replacing and training only the final layer. As training progressed, we gradually unfroze deeper layers to fine-tune feature extraction for soybean-specific patterns. The final fully connected layer was modified to output our four target classes. Training utilized a learning rate of 0.001 with Adam optimizer and implemented a maximum of 50 epochs with early stopping to prevent overfitting while enabling the model to learn the distinctive features of healthy and diseased soybean leaves.

DenseNet201: The DenseNet201 architecture, a deeper variant of DenseNet pretrained on ImageNet, was implemented for our soybean disease classification task. This network extends the dense connectivity pattern with additional layers, further enhancing feature propagation and reuse while maintaining parameter efficiency. We initially kept the base layers frozen while replacing and training only the classifier component. During training, we applied selective fine-tuning of deeper layers to optimize the model for soybean leaf image analysis. The final classification layer was modified to output our four target classes. Training was conducted using a learning rate of 0.001 with Adam optimizer and implemented a maximum of 50 epochs with early stopping to prevent overfitting while the model learned to differentiate between healthy and diseased soybean patterns. This suite of pretrained models allowed us to effectively transfer learned visual features and adapt them for the fine-grained classification of soybean conditions in UAV imagery.

5 Grad-CAM Visualization

Grad-CAM Algorithm and Purpose: Grad-CAM is a visualization method that highlights image regions influencing a model's prediction. It works by computing gradients of the target class with respect to feature maps in the last convolutional layer, producing heatmaps that show important areas. This helps confirm that the model focuses on disease-related features rather than irrelevant background.

Visualization Results and Interpretation: We employed three complementary Grad-CAM variants to visualize the regions our models focused on when making predictions. As shown in the visualization grid, Grad-CAM provided general attention maps highlighting key areas influencing model decisions across all four classes. Grad-CAM++ demonstrated improved localization capabilities with more concentrated hotspots, particularly evident in the center of the images where it identified disease-specific regions with fine-grained detail. Eigen-CAM, based on principal component analysis, produced broader activation patterns with distinct intensity gradients that offered a different perspective on feature importance. When comparing visualizations across classes (Healthy Soybean, Semilooper Pest Attack, Soybean Mosaic, and Rust), we observed that the models consistently focused on similar leaf features for diagnosis, suggesting the network identified fundamental visual patterns that differentiate healthy leaves from various disease conditions in UAV imagery, rather than relying on irrelevant background elements or artifacts shown in Figure3.

Observations by Class: Healthy Soyabean: Focus on uniform leaf structure and color, with the model attending to the characteristic healthy green tissue and consistent leaf

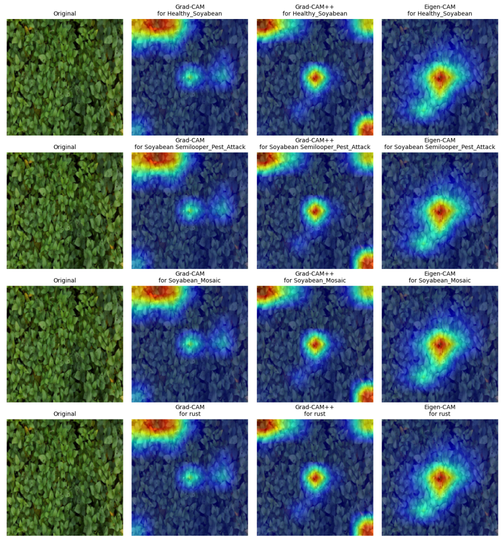

Figure 3: Grad-CAM, Grad-CAM++, and Eigen-CAM visualizations for soybean disease classification across four classes.

morphology.

Soyabean Semilooper Pest Attack: Highlights multiple pest-damaged areas, with concentrated activations on irregular feeding patterns and leaf perforations characteristic of insect damage.

Soyabean Mosaic: Strong activation on mottled yellow-green discoloration, with the model correctly identifying the irregular color patterns typical of viral infection.

Rust: Attention on reddish pustules and chlorotic tissue, with heat maps clearly emphasizing the distinctive rust-colored spots and surrounding affected leaf tissue.

These visualization results confirm that our models learned to identify disease-specific visual features rather than irrelevant background elements or artifacts in the UAV imagery.

Model Comparison: EfficientNet-B3, with the highest accuracy of 98.60%, showed the most precise focus on subtle disease textures and clear separation of similar features, with minimal background activation. This explains its superior performance compared to other models like DenseNet121 97.54% and DenseNet201/MobileNetV2 both 97.19%. The visualization analysis across models revealed that higher-performing architectures demonstrated more concentrated attention on disease-specific areas, while lower-performing ones like VGG16 34.39% accuracy showed more diffuse activation patterns. All top-performing models concentrated on meaningful disease features, supporting their practical use in UAV-based soybean monitoring systems.

6 Lime Analysis

LIME (Local Interpretable Model-agnostic Explanations) was implemented to provide interpretable explanations for soybean disease classification decisions across all trained models. The analysis works by generating 100 perturbed versions of each input image through selective masking of image regions (superpixels), then fitting a local linear model to identify the top 10 most influential regions that contribute to classification decisions for each disease category.

The LIME explanations reveal that models correctly focus on biologically relevant features: reddish-brown pustules for rust detection, mottled patterns for mosaic virus, and uniform green coloration for healthy leaves. This validation demonstrates that the deep learning models learned meaningful disease-specific features rather than spurious correlations, increasing confidence in the diagnostic system's reliability for agricultural applications.

6.1 visualization of Lime analysis

The LIME visualizations demonstrate (in Figure 4) effective model interpretability across all four soybean disease classes. The yellow-highlighted regions in the LIME explanations show the specific image areas that most influence the model's classification decisions.

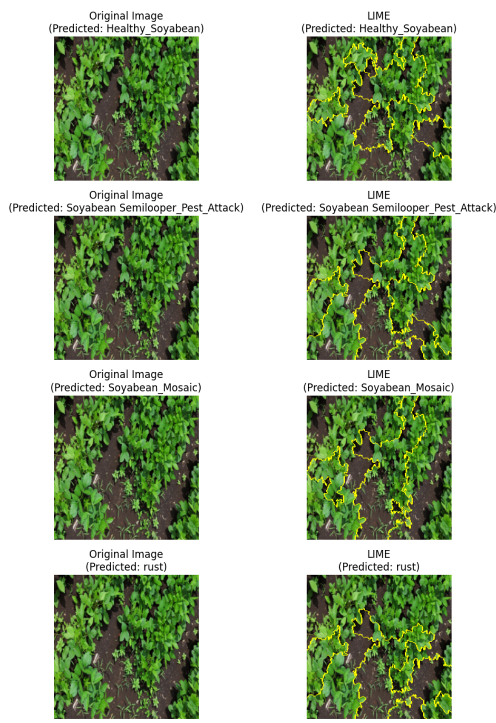

Figure 4: LIME explanations for soybean disease classification.

For **Healthy Soybea**n, LIME identifies uniform green leaf regions, validating the model's focus on healthy tissue characteristics. In **Semilooper Pest Attack** cases, the highlighted areas correspond to potential damage zones on leaf surfaces. For **Soybean Mosaic**, LIME emphasizes regions where mosaic patterns and discoloration typically appear. In **Rust** classification, the model correctly focuses on leaf areas prone to rust symptom development.

The consistent identification of biologically relevant leaf regions across all disease categories confirms that the model has learned meaningful disease-specific features rather than spurious correlations, providing confidence in the diagnostic system's reliability for practical agricultural applications.

7 Experimental Setup

7.1 Hardware and Software

7.1.1 Hardware

- **GPU:** NVIDIA GPU with CUDA support (Kaggle usually provides NVIDIA Tesla T4 or similar)
- **RAM:** Kaggle environments typically provide around 16 GB RAM

7.1.2 Software

- **Operating System:** Ubuntu Linux (default Kaggle environment)
- **Python Version:** 3.7+ (Kaggle default)
- **Key Libraries and Versions:**
  - `torch` (PyTorch) — used for deep learning models
  - `torchvision` — for pretrained models and data transforms
  - `numpy` — numerical computations
  - `matplotlib` — plotting
  - `scikit-learn` — evaluation metrics like precision, recall, f1-score
  - `pytorch-grad-cam` (installed via git) — for explainability
  - `lime` — for local interpretable model explanations
  - `tqdm` — progress bars
  - `PIL` (Pillow) — image processing

7.2 Training Details

The training process was executed using a batch size of 16 with the Adam optimizer (learning rate 0.001), incorporating early stopping with a patience of 5 epochs to prevent overfitting. Mixed-precision training (AMP) was implemented to accelerate the process and reduce memory consumption, with custom CNN models trained for up to 10 epochs and transfer learning models for a maximum of 50 epochs.

Early Stopping Strategy: Early stopping is implemented via a custom EarlyStopping class withPatience set to 5 epochs. After each epoch, validation loss is checked - if it decreases, the best loss is updated and counter resets; if it doesn't improve for 5 consecutive epochs, training stops early to prevent overfitting and save computational resources.

Model Saving: The model weights are saved after training completes or when early stopping triggers using torch.save(model.state_dict(), model_save_path). Different checkpoint files are saved per model type, e.g., "custom_cnn_model.pth", "transfer_learning_resnet50.pth", etc.

Table 4: Hyperparameters Used for Model Training

Hyperparameter	**Value**	**Notes**
Optimizer	Adam	Learning rate 0.001
Learning Rate	0.001	Constant (no explicit scheduler)
Batch Size	16	For train, validation, and test sets
Number of Epochs	10 for Custom CNN 50 for Transfer Learning models	Early stopping applied
Early Stopping	Yes	Patience = 5 epochs, monitors validation loss
Mixed Precision (AMP)	Enabled	For faster GPU training with less memory

7.3 Evaluation Metrics

- **Accuracy:** Measures the proportion of correctly classified samples over total samples.

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

- **Precision:** Measures how many predicted positives are actually positive.

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

- **Recall:** Measures how many actual positives were correctly predicted.

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

- **F1-Score:** Harmonic mean of precision and recall balancing both metrics.

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

Accuracy Threshold: An accuracy threshold above 80% is set because it ensures reliable disease detection which is critical to prevent crop loss, balances practical usability with achievable model complexity on real UAV image data, and indicates robust generalization on the unseen test data, critical for real-world agricultural deployment.

8 Results

Custom CNN Performance: Our custom CNN architecture achieved 89.47% accuracy, demonstrating its ability to learn disease-specific features directly from the UAV imagery. While it performed significantly below transfer learning models, it showed strong capabilities in identifying rust (F1-score 0.9741) and mosaic disease (F1-score 0.9178), but struggled

Table 5: Performance Metrics for Custom CNN on Test Set

Class	Precision	Recall	F1-Score
Healthy Soyabean	0.5714	0.4444	0.5000
Soyabean Semilooper Pest Attack	0.8542	0.9425	0.8962
Soyabean Mosaic	0.9178	0.9178	0.9178
Rust	0.9895	0.9592	0.9741
Weighted Average	**0.8902**	**0.8947**	**0.8910**
Overall Accuracy			**89.47%**

with healthy soybean classification (F1-score 0.5000), suggesting limitations in distinguishing subtle normal variation without pretrained knowledge, shown in Table-5

We implemented a custom CNN architecture with 6 convolutional layers and 5 fully connected layers for soybean disease classification from UAV imagery.

The confusion matrix in Figure5 demonstrates strong overall classification performance across the four soybean disease classes. The model achieves excellent accuracy for **Rust** detection with 94 correct predictions out of **98** samples, indicating robust identification of rust symptoms. **Soybean Semilooper Pest Attack** also shows strong performance with **82** correct classifications out of **87** total samples.

Soybean Mosaic classification achieves reasonable accuracy with 67 correct predictions out of **73** samples. However, **Healthy Soybean** classification shows some challenges, with only **12** correct predictions out of **27** samples, and notable confusion with Semilooper Pest Attack (13 misclassifications).

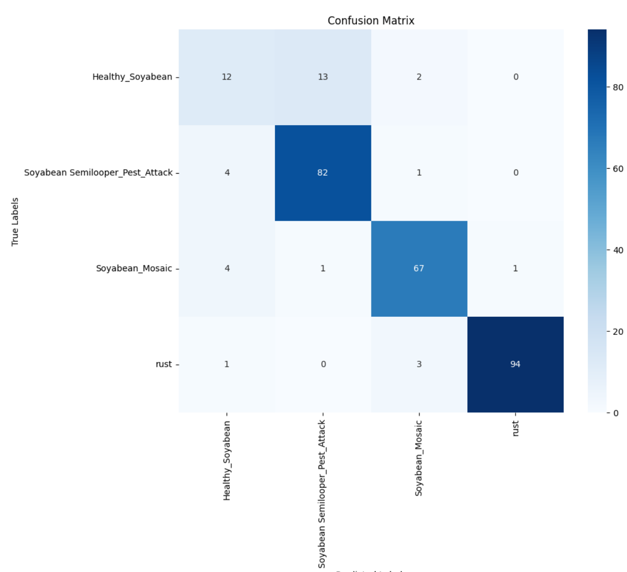

Figure 5: Confusion Matrix of custom cnn model.

The diagonal elements (dark blue regions) represent correct classifications, while off-diagonal elements indicate misclassifications. The matrix reveals that most confusion occurs between similar-appearing disease conditions, which is expected given the visual similarities in early-stage symptoms. Overall, the model demonstrates strong discriminative capability for disease detection in UAV-based soybean imagery.

The training and validation loss curves for the custom CNN, figure-6, demonstrate effective model learning over 21 epochs. Both losses start high (1.3) and rapidly decrease in the initial epochs, indicating quick convergence. The training loss (blue) shows smooth, consistent reduction throughout training. The validation loss (orange) exhibits more fluctuation with occasional spikes around epochs 6, 15, and 19, suggesting some instability during training. However, both curves converge to low values (0.2-0.4), indicating successful model optimization without significant overfitting. The parallel decrease of both losses demonstrates good generalization capability of the custom CNN architecture.

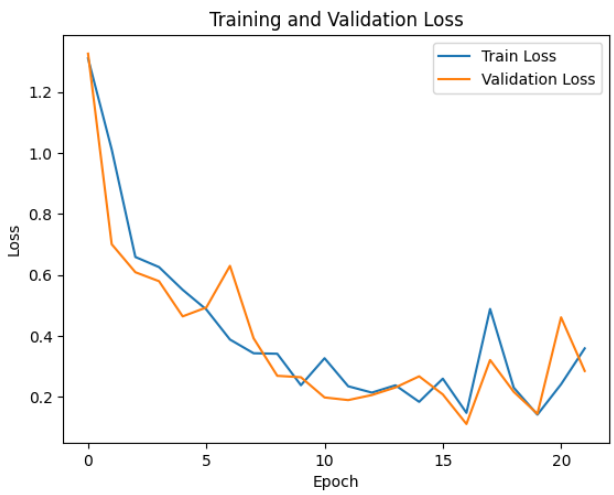

Figure 6: Training and validation loss curves for the custom CNN model.

We trained the model using Automatic Mixed Precision (AMP) for more efficient training while maintaining performance. The continuously decreasing validation loss suggests good generalization capability of our custom architecture, though subsequent experiments with transfer learning models achieved even higher performance for this specific task ; the performance is in table-6.

Table 6: Performance Metrics for Custom CNN with AMP on Test Set

Class	**Precision**	**Recall**	**F1-Score**
Healthy Soyabean	0.0000	0.0000	0.0000
Soyabean Semilooper Pest Attack	0.7477	0.9540	0.8384
Soyabean Mosaic	0.9365	0.8082	0.8676
Rust	0.8796	0.9694	0.9223
Weighted Average	**0.7706**	**0.8316**	**0.7953**
Overall Accuracy			**83.16%**

8.1 Transfer Learning Performance

Transfer learning proved highly effective for soybean disease classification, with pretrained models significantly outperforming the custom CNN approach. The knowledge transfer from ImageNet pretraining enabled models to leverage low-level feature extractors while adapting to specific soybean disease patterns, resulting in faster convergence and superior generalization, in table7.

Table 7: Performance Comparison of Models

Model	Accuracy (%)	Precision	Recall	F1-Score
ResNet-50	96.84	0.9684	0.9684	0.9684
VGG16	34.39	0.3439	0.3439	0.3439
MobileNetV2	97.19	0.9719	0.9719	0.9719
EfficientNet-B3	98.60	0.9860	0.9860	0.9860
DenseNet121	97.54	0.9754	0.9754	0.9754
ResNet152	95.44	0.9544	0.9544	0.9544
DenseNet201	97.19	0.9719	0.9719	0.9719

Here we got the best performing model and that is **EfficientNet-B3**(shown in table-7). The training and validation loss curves for EfficientNet-B3 demonstrate excellent convergence over 7 epochs with early stopping activation. The training loss (blue) shows smooth, consistent decrease from 0.29 to 0.07. The validation loss (orange) starts lower (0.15) and rapidly drops to 0.02 by epoch 2, remaining stable with minor fluctuations around epochs 4-6 before stabilizing at 0.04. The quick convergence indicates effective transfer learning, where the pre-trained EfficientNet-B3 features adapt rapidly to soybean disease classification. The validation loss remaining consistently low throughout training demonstrates excellent generalization without overfitting. Early stopping was triggered around epoch 7, preventing unnecessary training and ensuring optimal model performance. **Model**

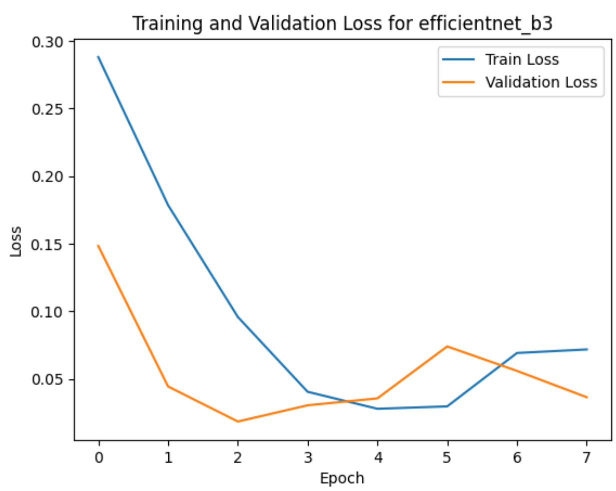

Figure 7: Training and validation loss curves for the best performing model.

Performance Analysis: The performance comparison reveals significant variance across the tested architectures. EfficientNet-B3 demonstrates superior performance with 98.6% accuracy, establishing it as the optimal model for UAV-based soybean disease classification. MobileNetV2 and DenseNet201 show strong results (both 97.19%), closely followed by DenseNet121 (97.54%) and ResNet50 (96.84%). ResNet152 performs slightly lower at 95.44%, while VGG16 shows substantially inferior performance (34.39%), likely due to its simpler architecture struggling with the subtle features in aerial imagery. The high performance of EfficientNet-B3 can be attributed to its efficient compound scaling design that balances network depth, width, and resolution, making it particularly effective at capturing the subtle visual patterns distinctive of various soybean diseases in UAV imagery,showed in Figure8.

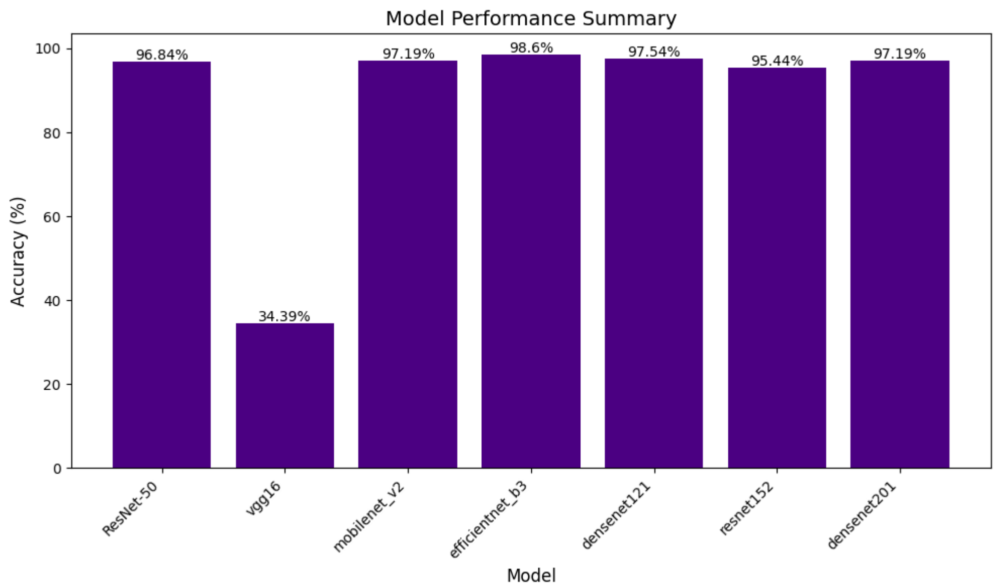

Figure 8: Training and validation loss curves for the best performing model.

8.2 Comparison and Discussion

Model Performance From the evaluation metrics in Table-7, we can conclude: **EfficientNet-B3** performed the best, with 98.60% accuracy, precision of 0.9860, recall of 0.9860, and F1-score of 0.9860. This strong performance is due to EfficientNet's balanced scaling, which effectively captures both fine and broad features in the images.

DenseNet121 achieved 97.54% accuracy, with its dense connectivity pattern enhancing feature reuse and improving gradient flow throughout the network.

MobileNetV2 and **DenseNet201** both reached 97.19% accuracy, demonstrating that efficient architectures can perform well in resource-limited environments, like drones or mobile devices. **VGG16** performed poorly with 34.39% accuracy and low precision (0.3439), recall (0.3439), and F1-score (0.3439). Its simpler architecture struggles to capture complex patterns needed for accurate disease classification from UAV imagery.

Impact of Preprocessing and Augmentation Our data preprocessing and augmentation strategies played a key role in the models' performance. Normalization helped standardize the input data, making training more stable across all architectures. Random resized cropping and horizontal flipping helped models adapt to changes in scale and orientation in UAV images, improving robustness to variable viewing angles and distances that are common in drone-captured imagery. The strong performance across most architectures indicates the effectiveness of our preprocessing approach in helping the models learn meaningful features that distinguish between different soybean conditions regardless of typical UAV image variations.

Practical Implications: EfficientNet-B3 is ideal for high-precision needs, like research or precision agriculture, where accuracy is crucial. MobileNetV2 offers strong performance and is lightweight, making it suitable for on-device applications like drones in the field. The overall high performance (except VGG16) suggests that deep learning can be successfully used for soybean disease detection in real-world agricultural settings.

These findings guide model selection for UAV-based disease detection in agriculture.

9 Discussion

Trade-offs: Accuracy vs. Computation Time and Model Size: Our analysis revealed important trade-offs between performance and efficiency for UAV-based soybean disease detection. EfficientNet-B3 achieved the highest accuracy (98.60%) but requires more computational resources for training and inference. MobileNetV2 delivered impressive results (97.19% accuracy) despite having significantly fewer parameters, making it ideal for resource-constrained environments like drone-mounted systems or edge devices. DenseNet variants offered excellent performance with moderate computational demands, while VGG16's poor performance (34.39% accuracy) despite its large parameter count demonstrates that architectural design is more critical than model size for this task. These findings suggest that model selection should balance accuracy requirements against the practical constraints of deployment in agricultural settings.

Surprising Findings and Limitations: The strong performance of lightweight architectures like MobileNetV2 (97.19% accuracy) contradicts the conventional assumption that larger models generally perform better, suggesting efficient design can effectively capture disease patterns in UAV imagery. VGG16's significant under-performance (34.39%) highlights that simpler architectures struggle with the complex patterns in aerial agricultural imagery, emphasizing the importance of modern architectural innovations for this domain. While our models showed good generalization on the test set, the limited dataset size remains a consideration for real-world deployment where more variable field conditions may be encountered. Additionally, all models were trained on 224×224 pixel images, potentially limiting the detection of fine-grained disease features that might be visible at higher resolutions.

For practical deployment, the choice between EfficientNet-B3 and other high-performing models should be guided by specific use-case requirements, balancing accuracy advantages against computational efficiency needs in agricultural monitoring systems.

The strong performance of lightweight architectures like MobileNetV2 (97.19% accuracy) contradicts the conventional assumption that larger models generally perform better, suggesting efficient design can effectively capture disease patterns in UAV imagery. VGG16's significant underperformance (34.39%) highlights that simpler architectures struggle with the complex patterns in aerial agricultural imagery, emphasizing the importance of modern architectural innovations for this domain. While our models showed good generalization on the test set, the limited dataset size remains a consideration for real-world deployment where more variable field conditions may be encountered. Additionally, all models were trained on 224×224 pixel images, potentially limiting the detection of fine-grained disease features that might be visible at higher resolutions.

For practical deployment, the choice between EfficientNet-B3 and other high-performing models should be guided by specific use-case requirements, balancing accuracy advantages against computational efficiency needs in agricultural monitoring systems.

10 Conclusion

This study successfully implemented and evaluated seven transfer learning models for UAV-based soybean disease classification: ResNet-50, VGG16, MobileNetV2, EfficientNet-B3, DenseNet121, ResNet152, and DenseNet201. Among these: Six models achieved test accuracies above 95%,validating the effectiveness of transfer learning for plant disease detection in real-world imagery. The EfficientNet-B3 model outperformed all others, achieving the highest test accuracy of 98.60%. DenseNet121 showed excellent performance with 97.54% accuracy, while MobileNetV2 and DenseNet201 both reached 97.19%, maintaining relatively fast training times and lower computational complexity. Even models like ResNet-50 (96.84%) and ResNet152 (95.44%) performed exceptionally well. Only VGG16 fell significantly short, with a test accuracy of 34.39%, indicating its limitations for this specific classification task.

Based on performance metrics and practical considerations such as training efficiency and model complexity, the following recommendations are made: EfficientNet-B3 is recommended as the primary model due to its superior accuracy (98.60%), efficient architecture, and robust generalization across disease classes. For environments with limited computational resources, MobileNetV2 is a suitable alternative, offering a good trade-off between performance (97.19%) and efficiency with its lightweight design. DenseNet121 (97.54%) presents another excellent option that balances accuracy and computational demands. Models such as VGG16 are not recommended due to significantly lower accuracy (34.39%) and heavier architecture, which make them less effective for soybean disease classification using UAV imagery.

Interpretability and Explainable AI Insights enhances model transparency, multiple explainability techniques were implemented including Grad-CAM, Grad-CAM++, Eigen-CAM, and LIME. These visualization methods highlighted the specific leaf regions that contributed most to the model's decision-making, confirming that models focused on biologically relevant disease patterns rather than background noise. For healthy soybean, the models attended to uniform leaf structure, while for diseases, they highlighted distinctive features - insect damage patterns for semilooper pest attack, mottled discoloration for mosaic disease, and characteristic pustules for rust infection.

LIME analysis further validated these findings by generating local explanations through superpixel-based perturbations, demonstrating consistent focus on disease-relevant leaf regions across all model architectures.The complementary nature of gradient-based (Grad-

CAM family) and perturbation-based (LIME) explanations provided comprehensive model interpretability, critical for real-world adoption, allowing agronomists and end-users to verify and trust model predictions, thereby enhancing the credibility and practical value of the deep learning systems for UAV-based soybean disease monitoring.

11 Future Work

Dataset expansion: Collect UAV imagery across different growing seasons and regions to improve model robustness to environmental variations. Multi-spectral imaging: Incorporate near-infrared or thermal imagery to potentially detect diseases before visible symptoms appear. Temporal monitoring: Develop systems to track disease progression over time, enabling early intervention strategies. Model optimization: Investigate quantization and pruning techniques to improve deployment efficiency on drone-mounted edge devices. Disease severity quantification: Extend classification to estimate disease severity percentages, providing more actionable information for precision agriculture. Ensemble methods: Combine predictions from complementary models to potentially improve accuracy while maintaining reasonable inference times. These improvements would advance UAV-based soybean disease detection toward more effective implementation in real-world agricultural settings.

References

[1] Wu, Q., Ma, X., Liu, H., Bi, C., Yu, H., Liang, M., Zhang, J., Li, Q., Tang, Y. and Ye, G., 2023. A classification method for soybean leaf diseases based on an improved ConvNeXt model. *Scientific Reports*, 13(1), p.19141.

[2] Yu, M., Ma, X., Guan, H., Liu, M. and Zhang, T., 2022. A recognition method of soybean leaf diseases based on an improved deep learning model. *Frontiers in plant science*, 13, p.878834.

[3] Rodríguez-Lira, D.C., Córdova-Esparza, D.M., Álvarez-Alvarado, J.M., Romero-González, J.A., Terven, J. and Rodríguez-Reséndiz, J., 2024. Comparative Analysis of YOLO Models for Bean Leaf Disease Detection in Natural Environments. *AgriEngineering*, 6(4), pp.4585-4603.

[4] Goshika, S., Meksem, K., Ahmed, K.R. and Lakhssassi, N., 2023. Deep learning model for classifying and evaluating soybean leaf disease damage. *International Journal of Molecular Sciences*, 25(1), p.106.

[5] Slimani, H., El Mhamdi, J. and Jilbab, A., 2023. Drone-assisted plant disease identification using artificial intelligence: A critical review. *International Journal of Computing and Digital Systems*, 14(1), pp.10433-10446.

[6] Bhargava, A., Shukla, A., Goswami, O.P., Alsharif, M.H., Uthansakul, P. and Uthansakul, M., 2024. Plant leaf disease detection, classification, and diagnosis using computer vision and artificial intelligence: A review. *IEEE access*, 12, pp.37443-37469.

[7] Jahin, M.A., Shahriar, S., Mridha, M.F., Hossen, M.J. and Dey, N., 2025. Soybean Disease Detection via Interpretable Hybrid CNN-GNN: Integrating MobileNetV2 and GraphSAGE with Cross-Modal Attention. *arXiv preprint arXiv:2503.01284*.

[8] Al Sahili, Z. and Awad, M., 2022. The power of transfer learning in agricultural applications: AgriNet. *Frontiers in Plant Science*, 13, p.992700.

A Appendix: Hyperparameters and Code Snippets

Appendix: Hyperparameters and Code Snippets (shown in Table 8)

This section summarizes the key hyperparameters used across different models and training procedures. Table 8 details the common settings, data augmentation techniques, and model-specific configurations essential for reproducing our experiments.

Table 8: Model Hyperparameters

Parameter	**Value/Details**
Common Settings	
Batch Size	16
Learning Rate	0.001
Optimizer	Adam
Loss Function	CrossEntropyLoss
Early Stopping Patience	5
Image Resolution	224×224
Normalization	Mean=0.5, Std=0.5
Data Augmentation	
Training Set	Random Resized Crop (224×224), Random Horizontal Flip
Validation/Test Set	Center Resize (224×224)
Model-Specific	
Custom CNN	6 conv layers, 5 FC layers, Dropout rate=0.1
ResNet-50	Final FC: 2048→4 features
MobileNetV2	Final Classifier: 1280→4 features
EfficientNet-B3	Final Classifier: 1536→4 features
DenseNet121	Final Classifier: 1024→4 features