

Tariq et al. LGURJCSIT 2025

ISSN: 2521-0122 (Online)
ISSN: 2519-7991 (Print)

LGU Research Journal of Computer Science & IT

doi: 10.54692/lgurjcsit.2025.092725

Vol (9): Issue (2), July – December 2025

Automated Tomato Disease Diagnosis Using Deep Learning: A Robust Approach with XAI and Transfer Learning

Muhammad Usman Tariq[1], Aamir Hussain[1*], Salman Qadri[1], Sarfraz Hashim[2], Altaf Hussain[1], Hafiz Nazar Faried[3], Rashda[1], Shahid Zaman[1]

[1]Institute of Computing, Muhammad Nawaz Shareef University of Agriculture, Multan, Pakistan.

[2]Department of Agricultural Engineering, Muhammad Nawaz Shareef University of Agriculture, Multan, Pakistan

[3]Department of Horticulture, Muhammad Nawaz Shareef University of Agriculture, Multan, Pakistan.

Email: aamir.hussain@mnsuam.edu.pk

ABSTRACT:

Tomato cultivation is a vital component of global agriculture, yet its productivity is severely constrained by a wide range of leaf diseases that negatively impact both yield and market value. Conventional methods of disease identification, which rely on manual inspections and laboratory testing, are slow, resource-intensive, and prone to human error. This study proposes a robust deep learning-based framework for the automated diagnosis of tomato leaf diseases, integrating transfer learning and explainable artificial intelligence (XAI) for improved accuracy and transparency. The model leverages EfficientNetV2B0 as the backbone architecture, enhanced with a custom channel attention mechanism and fine-tuning strategies to improve feature extraction and classification. Data augmentation and hyperparameter optimization were applied to address dataset imbalance and noise. The system achieved high classification accuracy across multiple disease classes while maintaining generalizability. Furthermore, Grad-CAM and saliency maps were employed to provide visual explanations of the model's predictions, thereby enhancing interpretability and trust in real-world applications. Experimental results demonstrate that the proposed method significantly improves diagnostic precision and reliability compared to traditional approaches, offering a scalable and practical solution for early detection of tomato leaf diseases in modern agriculture.

KEYWORDS: Tomato Leaf Disease, Deep Learning, Transfer Learning, Convolutional Neural Networks, Explainable AI, Grad-CAM, Plant Disease Detection

1. INTRODUCTION

Solanum lycopersicum, commonly known as tomato, is among the most widely cultivated vegetable crops across the globe because of its nutritional richness and economic importance. Tomatoes significantly contribute to food security, income generation, and employment, particularly in developing regions where horticulture remains a pillar of the rural economy. The fruit serves multiple consumption purposes fresh, processed, or industrially packaged making it one of the most traded commodities worldwide. Its nutritive profile, rich in vitamins A, C, and K, essential minerals, and potent antioxidants such as lycopene, further enhances its significance for human health and wellbeing (Tiwari et al., 2022; Abu Haraira et al., 2022). Despite these benefits, tomato production is increasingly threatened by several biotic and abiotic stressors that substantially reduce crop productivity. Fungal, bacterial, and viral diseases such as early blight, late blight, bacterial spot, mosaic virus, and tomato yellow leaf curl virus can cause yield losses ranging from 20% to over

100% in severe cases (Gatahi, 2020; Prasad et al., 2020). In addition, non-biological stressors such as nutrient deficiency, heat, and water stress further compromise plant health and productivity (Martínez-Dalmau et al., 2021). These challenges severely impact farmers' livelihoods and the overall sustainability of agricultural systems. Conventional methods for disease detection rely on manual field inspection and laboratory analysis, which are not only labor-intensive and time-consuming but also prone to human error and inconsistency. As global food demands rise, automation in agricultural diagnostics has become a necessity. Recent advancements in artificial intelligence (AI) and computer vision have enabled a paradigm shift toward precision agriculture. Specifically, deep learning and Convolutional Neural Networks (CNNs) have shown exceptional performance in image-based plant disease classification by automatically learning discriminative features from visual data without manual feature engineering (A. Khan et al., 2020; Shaikh et al., 2022). These approaches have outperformed traditional machine learning techniques in both speed and accuracy, making them suitable for real-time and large-scale agricultural monitoring. However, agricultural datasets are often limited in size and diversity, posing challenges to model generalization. To overcome this, transfer learning has emerged as an effective strategy that utilizes pre-trained models such as EfficientNet, ResNet, and VGG, enabling high performance with reduced computational cost and training data requirements (Ngugi et al., 2024; Farman et al., 2022). Although deep learning has achieved remarkable accuracy, its black-box nature often limits trust and adoption among end-users, particularly in the agricultural domain where interpretability is critical. Explainable Artificial Intelligence (XAI) addresses this issue by providing interpretability and visual transparency to AI-driven predictions. Techniques like Gradient-weighted Class Activation Mapping (Grad-CAM) and saliency maps help highlight discriminative leaf regions influencing model predictions, thus improving accountability, trust, and practical usability (Ahmed et al., 2025). The integration of XAI with deep learning ensures that automated predictions are transparent and communicable, making advanced models more accessible to farmers and agronomists. The present research proposes a robust and interpretable tomato leaf disease diagnosis framework that integrates Efficient NetV2B0 with custom attention mechanisms and visual explanation tools. This model aims to address three key challenges — dataset imbalance, diagnostic reliability, and interpretability — thereby contributing toward a scalable, transparent, and efficient AI system for precision agriculture. Ultimately, this approach fosters early disease detection, informed intervention, and sustainable crop management practices that align with the real-world needs of modern agriculture.

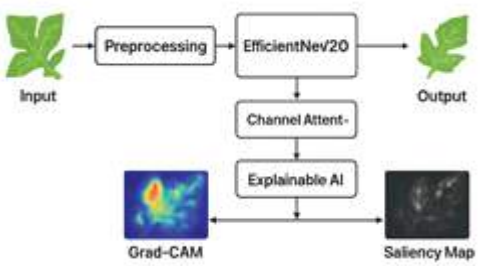

Figure 1: Proposed framework for tomato leaf disease detection and classification

The overall workflow of the proposed framework is presented in Figure X, which summarizes the sequence of major components from input to explainable output.

2. LITERATURE REVIEW

Plant disease detection has long been a central problem in precision agriculture, with early research relying mainly on traditional image-processing and classical machine learning techniques. Early models such as Support Vector Machines (SVMs), k-Nearest Neighbours (k-NN), and Random Forests were trained using manually engineered features like color histograms, texture, and shape descriptors (Shaikh et al., 2022). Although these methods performed well in controlled environments, their generalization capability was limited due to sensitivity to variations in lighting, background, and leaf morphology (Krishna & Kalluri, 2019). These constraints made them less suitable for large-scale or real-time applications in diverse agricultural conditions.

The emergence of Deep Learning (DL), particularly Convolutional Neural Networks (CNNs), marked a transformative shift. CNNs automatically learn hierarchical representations directly from raw image pixels, eliminating the need for manual feature engineering (A. Khan et al., 2020). Early CNN-based models achieved over 90% classification accuracy on benchmark datasets such as

PlantVillage, demonstrating their potential for disease detection in tomato and other crops. However, architectures like AlexNet and VGG16, though accurate, involved millions of parameters and high computational costs, making them inefficient for real-time deployment in agricultural settings (Howard et al., 2017).

To overcome these limitations, researchers explored ResNet and Inception architectures, which improved gradient flow and multi-scale feature extraction (He et al., 2016). The introduction of lightweight CNN architectures such as MobileNet and EfficientNet brought a balance between model efficiency and accuracy, allowing edge deployment on devices with limited computational resources (Tan & Le, 2019; Krishna & Kalluri, 2019). Notably, EfficientNetB0 achieved high accuracy with significantly fewer parameters compared to VGG16 and ResNet50, making it an ideal base for agricultural diagnostic models where computational efficiency is critical. Another major advancement came through Transfer Learning, which allows pre-trained models (e.g., EfficientNet, ResNet, DenseNet) trained on large datasets like ImageNet to be fine-tuned for domain-specific agricultural datasets. This approach significantly reduces training time and improves performance, especially in data-scarce agricultural domains (Ngugi et al., 2024; Farman et al., 2022). However, despite their effectiveness, these models are still limited by their computational demand and lack of interpretability—two key concerns for field deployment in rural farming contexts. To improve performance further, attention mechanisms have been introduced into CNN architectures. Channel attention modules emphasize the most informative feature maps, while spatial attention highlights important image regions, allowing the model to focus on disease-relevant areas (Sharma & Shivandu, 2024). This mechanism improves classification sensitivity, especially when distinguishing visually similar diseases such as late blight and leaf mold. Nevertheless, these improvements often come at the cost of increased computational complexity, which has led to the exploration of integrating attention into lightweight architectures like EfficientNetV2B0 — the foundation model used in this study. A persistent gap in the literature is the lack of interpretability in deep learning-based diagnostic systems. While CNNs and attention mechanisms can yield high accuracy, they often function as "black boxes," providing limited insight into how predictions are made (M. Khan et al., 2024). In the agricultural domain, where users are primarily farmers and agronomists, interpretability is critical to trust and adoption. Explainable Artificial Intelligence (XAI) methods, such as Grad-CAM, Saliency Maps, LIME, and SHAP, have emerged to visualize model reasoning by identifying key regions influencing predictions (Ngugi et al., 2024; Ahmed et al., 2025). These methods bridge the gap between machine learning outputs and human understanding, ensuring accountability and practical usability. However, most prior works employ XAI merely as a post-hoc visualization step, rather than integrating it into the diagnostic workflow. Moreover, challenges such as dataset noise, varying lighting conditions, and overlapping disease symptoms continue to affect classification reliability. Thus, there remains a clear need for a holistic approach that balances accuracy, computational feasibility, and interpretability. Building upon these research gaps, the current study introduces a hybrid framework combining EfficientNetV2B0 with a custom Channel Attention Activator and integrated XAI modules (Grad-CAM and Saliency Maps). This combination ensures that the model not only achieves high classification accuracy but also provides transparent and interpretable visual justifications for its predictions. By addressing the three major challenges — dataset imbalance, reliability, and explainability — this research extends the existing literature toward a more trustworthy, lightweight, and field-ready AI-based tomato disease detection system.

3. DATASET

The dataset used in this study was carefully curated from PlantVillage resources and supplementary agricultural image collections, ensuring representation from both controlled experimental environments and real-field conditions. This combination enhanced model generalization and robustness against environmental variations such as lighting, background noise, and leaf orientation. The final dataset comprised 22,930 tomato leaf images, categorized into ten classes — nine diseased and one healthy. For this research, we particularly focused on biotic tomato leaf diseases, including.

a) Early blight
b) Late blight
c) Leaf mold
d) Bacterial spot
e) Septoria leaf spot

f) Mosaic virus
g) Target spot

To ensure the dataset's quality and fairness, clear inclusion and exclusion criteria were applied.

- Inclusion criteria: images showing clear visual symptoms of the targeted diseases, verified through expert reference or PlantVillage annotations, and captured under adequate illumination.

- Exclusion criteria: images with excessive background clutter, blurred focus, multiple overlapping leaves, or uncertain labeling were removed during the curation phase.

Data imbalance across disease categories (e.g., more samples for Late Blight and Bacterial Spot than Mosaic Virus or Leaf Mold) was mitigated through data augmentation techniques, including:

- Rotation: up to ±25° to simulate natural variations in leaf orientation.
- Width and Height Shifts: up to 15% of the image dimensions to account for off-centered captures.
- Zoom Range: between 0.8× and 1.2× to replicate varying camera distances.
- Shear Intensity: up to 0.2 radians to introduce mild geometric distortion.
- Horizontal and Vertical Flips: to mimic leaves appearing in different orientations.
- Brightness Adjustment: ranging from 0.8 to 1.2 to simulate field lighting variability.

improved the model's ability to generalize to unseen conditions.

The dataset was divided into three subsets:

a) Late Training set (70%): Used for model learning and weight optimization.
b) Validation set (15%): Used for hyperparameter tuning and overfitting prevention.
c) Test set (15%): Unseen data reserved for final evaluation of the model's generalization.

dditionally, a data cleaning pipeline was applied using Python and TensorFlow utilities to remove duplicate or corrupted files and normalize image dimensions (224×224 pixels). All preprocessing steps were performed uniformly to ensure reproducibility. Through this systematic construction, cleaning, and augmentation process, the dataset now fully supports a transparent and reproducible framework for tomato leaf disease diagnosis using EfficientNetV2B0 with integrated XAI. The dataset utilized was primarily sourced from the PlantVillage open-access repository, which is publicly available for academic and research purposes under a Creative Commons Attribution (CC BY) license.

4. METHODOLOGY

The paper combines deep learning, transfer learning, attention, and explainable artificial intelligence (XAI) into one model that automatically recognizes tomato leaf diseases. The system comprises a step history of pipeline consisting of collecting and preprocessing datasets and extracting features based on the pretrained pipeline by a backbone network, enriching them with a channel attention module, and classifying them through a modified structure of dense layers. To achieve transparency and reliability, the XAI methods of Grad-CAM and saliency maps are included and used to demonstrate how the model has decided. The end-to-end processing is expected to accomplish three goals: (1) disease classification in a variety of tomato leaf conditions correctly, (2) computational efficiency with irony problems in real-life agricultural settings, and (3) interpretability that instills confidence in the forecasts of the system. The given framework focuses on the EfficientNetV2B0 backbone, which is reputable due to the use of fewer parameters and its compatibility with scale. Efficient Net uses a compound scaling principle to uniformly balance the dimensions depth, width, and resolution, unlike traditional CNNs that scale each separately. This systematically grows the network, resulting in improved performance using fewer parameters. Specifically, EfficientNetV2 also adds fused-MBConv blocks, which combine convolution and expansion in a single block, in the initial layers, speed up the training process, and enhance the low-level features. It further maintains the squeeze-and-excitation (SE) modules part, which are the feature responses that are adaptively adjusted across channels. EfficientNetV2B0 was selected as it has the best trade-off between the accuracy and its computational cost compared to larger ones (B1, B7). Although more accuracy can be achieved in the deeper versions, there is a lot more in the deep version, which can be problematic when implemented in low-resource agricultural environments. In comparison, B0 can have a high level of accuracy with several parameters at around 7 million. In contrast, in VGG16 and ResNet50, the number of parameters is at around 138 million and 25 million, respectively, and serve better in real-time applications as well as edge devices. To complete the activity of discrimination power, a custom channel attention module was integrated in the network. Although Efficient Net already

contains SE blocks, a separate channel attention mechanism gives a better opportunity to regulate the significance of feature maps. The mathematical model of channel paying attention can be mathematically stated as: The channel attention mechanism can be expressed as,

$M(F)=DC(\sigma(W2\delta(W1GAP(F))))$

GAP(F) represents the global average pooling operation applied to the feature map F. The matrices W1 and W2 are learnable weights of the gating network, while σ( ) denotes the ReLU and sigmoid activation functions, respectively.

This process generates the channel attention map Mc(F), which is then used to reweight the original feature maps. By doing so, the network emphasizes channels that carry disease-specific information, enhancing the model's ability to focus on the most relevant features during diagnosis.

This mechanism is particularly useful in tomato disease classification, where certain conditions exhibit subtle differences in lesion patterns. For example, early blight and target spot manifest as necrotic spots, but attention-enhanced features allow the model to focus on lesion borders and color gradients, improving class separability.

The complete architecture proceeds as follows:

- Input Layer: Preprocessed RGB images of size 224 × 224 × 3.
- EfficientNetV2B0 Backbone: Extracts hierarchical visual features using fused-MBConv and SE modules. Lower layers remain frozen to preserve general representations, while upper layers are fine-tuned.
- Channel Attention Module: Applied after selected convolutional blocks to reweight channel-wise features. This improves sensitivity to critical disease-specific patterns while reducing noise from irrelevant leaf textures or background.
- Global Average Pooling (GAP): Reduces high-dimensional feature maps into compact feature vectors, lowering the risk of overfitting.
- Fully Connected Layers: A dense layer of 512 neurons with ReLU activation, a dropout layer (rate = 0.5) for regularization, and a second dense layer of 128 neurons for refined feature representation.
- Output Layer: SoftMax activation outputs probability scores across 10 classes (nine diseased, one healthy).

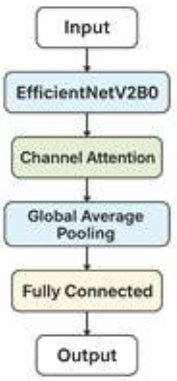

Figure 2: Training and validation accuracy

This design ensures robust classification while maintaining efficiency. Combining EfficientNetV2B0's compound scaling, SE modules, and the custom channel attention block enables the framework to outperform traditional CNN baselines in accuracy and interpretability. Moreover, the relatively lightweight structure makes the system practical for field-level deployment, including mobile devices and edge computing platforms.

4.1. Training Strategy

The training phase was carefully designed to ensure robust learning and prevent overfitting, a common challenge in agricultural datasets due to class imbalance and noise. The following strategies were implemented.

A. Training History And Visual Tracking

During model training, an important safeguard was put in place: a callback that automatically saved the best version of the model. Instead of keeping every update (even the bad ones), the system monitored the model's performance on a validation set at the end of each epoch. Whenever the model improved, the updated weights were stored. Those weaker updates were ignored if performance dipped in later epochs—often due to overfitting. This ensured we always preserved the model's strongest, most accurate version. Think of it as keeping backup saves in a video game—you only hang onto the runs where you performed best.

The final saved model was then used for testing, guaranteeing that reported results reflected peak performance in terms of both accuracy and loss. These stored weights also became a reliable

reference for further experiments, such as interpretability studies using techniques like Grad-CAM, or simply as a safety net if training progress wavered. Beyond checkpointing, the training process was also adapted dynamically thanks to the ReduceLROnPlateau callback. This mechanism continuously monitors the validation accuracy. If the accuracy stopped improving for three epochs, the learning rate was automatically reduced by 0.2. Lowering the learning rate at these stalls allowed the optimizer to make more minor, more careful adjustments—beneficial toward the later stages of training, when the model was fine-tuning its performance. A minimum learning rate was also enforced to keep things stable, so the optimizer never ground to a complete halt.

This careful combination—learning rate scheduling and Adam's adaptive gradient updates—played a significant role in keeping training smooth and stable, especially during the delicate fine-tuning phase when both the EfficientNetV2B0 backbone and the custom channel attention layers had to work harmoniously.



Figure 3: Training and validation accuracy

Figure 3 illustrates how training and validation accuracy evolved across epochs.

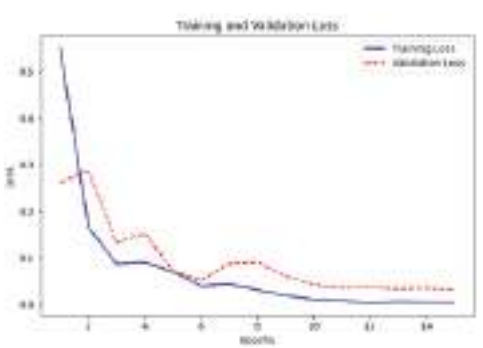

Figure 4: Training and validation loss

Figure 4 charts the corresponding loss curves. Together, they provide a clear view.

B. Optimizer

The Adam optimizer was employed because of its adaptive learning rate mechanism that combines the advantages of AdaGrad and RMSProp. It automatically adjusts the learning rate for each parameter, improving convergence speed. An initial learning rate of 0.001 was selected after experimental tuning. A learning rate scheduler was implemented to further refine optimization, which reduced the learning rate by a factor of 0.1 when validation accuracy plateaued for five consecutive epochs. This dynamic adjustment prevented stagnation and improved final convergence.

C. Loss Function

The classification task was framed as a multi-class problem; therefore, categorical cross-entropy was used as the objective function. This loss is well-suited for probability distribution outputs and penalizes incorrect predictions based on the divergence between predicted and true labels.

The loss function can be expressed as:

$$L = -\Sigma \text{ (from } i=1 \text{ to } C) \left[y_i \cdot \log(\hat{y}_i) \right] \quad (1)$$

Where:

- C is the number of classes.
- yi represents the ground truth label.
- y^i denotes the predicted probability for class i.

D. Batch Size And Epochs

A batch size of 32 was used, balancing stability of gradient updates with memory efficiency. Training was conducted for a maximum of 50 epochs, but an early stopping criterion with a patience of 7 epochs was applied. This halted training once validation performance ceased to improve, preventing overfitting and saving computation time.

E. Regularization Methods

To further reduce overfitting risks, multiple regularization techniques were employed:

- Dropout layers (rate = 0.5) were integrated into the fully connected layers to deactivate neurons during training randomly.
- L2 weight decay regularization was applied to penalize large weights and encourage smoother generalization.

• Early stopping acted as an additional safeguard by monitoring validation loss time.

F. Hyperparameter Tuning

Hyperparameters, including the learning rate, batch size, dropout rate, and optimizer type, were systematically optimized through a grid search strategy. Initial experiments compared the performance of Adam, SGD with momentum, and RMSProp optimizers, with Adam demonstrating the best trade-off between accuracy, stability, and convergence speed. The final configuration employed the Adam optimizer with a learning rate of 0.001, an adaptive learning rate scheduler, a batch size of 32, and a dropout rate of 0.5. This configuration consistently achieved the highest validation accuracy and prevented overfitting across multiple runs.

G. Regularization Methods

To further reduce overfitting risks, multiple regularization techniques were employed:

• Dropout layers (rate = 0.5) were integrated into the fully connected layers to deactivate neurons during training randomly.

• L2 weight decay regularization was applied to penalize large weights and encourage smoother generalization.

• Early stopping acted as an additional safeguard by monitoring validation loss time.

H. Hyperparameter Tuning

Hyperparameters such as learning rate, batch size, dropout rate, and optimizer type were tuned through a grid search approach. Preliminary experiments compared Adam, SGD with momentum, and RMSProp optimizers, with Adam providing the best accuracy and convergence speed balance. The final configuration included Adam, a batch size of 32, dropout of 0.5, and a learning rate of 0.001 with adaptive scheduling.

I. Hardware And Software Environment

The experiments were conducted using a GPU-enabled environment. The framework used was TensorFlow 2.x with Keras' API, running on Python 3.8. Google Colab was utilized for initial prototyping and execution due to its accessibility and built-in GPU support.

J. Evaluation Metrics

Multiple evaluation metrics were employed beyond simple accuracy to comprehensively assess the proposed model's performance. These metrics provide insights into both overall and class-specific performance:

Accuracy: Defined as the proportion of correctly classified samples to the total number of samples. While accuracy provides a global performance measure, it can be misleading under class imbalance conditions.

$$\text{Accuracy} = (TP + TN) \div (TP + TN + FP + FN) \quad (2)$$

• Precision, Recall, and F1-Score: These metrics were calculated for each class and averaged using a macro-weighted scheme:

• Precision evaluates how many predicted positive cases are truly positive.

• Recall (Sensitivity) measures how many actual positive cases were correctly identified.

• F1-Score is the harmonic means of precision and recall, providing a balanced metric in imbalanced datasets

$$\text{Precision} = TP \div (TP + FP) \quad (3)$$

$$\text{Recall} = TP \div (TP + FN) \quad (4)$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) \div (\text{Precision} + \text{Recall}) \quad (5)$$

	precision	recall	f1-score	support
Tomato__Early_blight	0.98	0.99	0.98	223
Tomato__Late_blight	1.00	0.99	0.99	286
Tomato__Leaf_Mold	0.99	1.00	1.00	315
Tomato__Septoria_leaf_spot	1.00	0.98	0.99	306
Tomato__Target_Spot	0.99	0.99	0.99	281
Tomato__Tomato_mosaic_virus	0.96	1.00	0.98	45
Tomato__healthy	0.99	0.99	0.99	191
accuracy			0.99	1647
macro avg	0.99	0.99	0.99	1647
weighted avg	0.99	0.99	0.99	1647

Figure 4: Classification report

K. Confusion Matrix

A class-wise confusion matrix was generated to visualize correct and incorrect classifications. This helped identify specific classes (e.g., late blight vs. target spot) where the model struggled due to visual similarity.

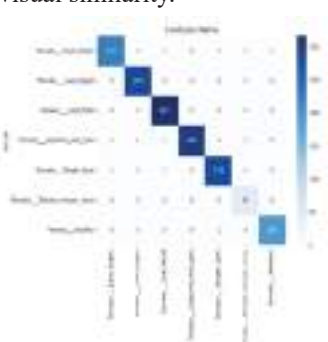

Figure 6: Confusion Metrix

L. Explainable AI Integrations

Although deep learning models achieve remarkable accuracy in image classification tasks, their black-box nature limits transparency and trust in real-world agricultural applications. Farmers, agronomists, and decision-makers require accurate predictions and clear reasoning. The proposed framework incorporates Explainable Artificial Intelligence (XAI) techniques to address this limitation, enabling visualization and interpretation of the features that drive model decisions. This study applied two complementary methods: Gradient-weighted Class Activation Mapping (Grad-CAM) and Saliency Maps.

- ***Grad-Cam***

Grad-CAM generates a class-specific localization map that highlights the regions of an image most responsible for the model's classification decision. It works by tracing the gradient information of a particular class back through the final convolutional layer, which allows it to assign varying levels of importance to different feature maps. This process results in a heatmap emphasizing the area's most relevant to the prediction. The ReLU function ensures that only the positively contributing features are retained. The resulting heatmap is typically superimposed on the original image, visually explaining which regions influenced the classification. For example, in tomato disease detection, Grad-CAM may highlight dark concentric rings in cases of early blight, or focus on curled leaf margins and vein areas in the case of yellow leaf curl virus. This makes it a powerful interpretability tool for identifying disease-specific visual cues.



Figure 7: Explainable AI Visualization

M. Saliency Maps

In contrast to Grad-CAM, which offers a broader, coarse-level visualization, saliency maps provide pixel-level insights into model sensitivity. They indicate how small changes in each input image pixel can influence the classification outcome. Brighter areas in the saliency map reveal regions where the prediction is particularly sensitive to pixel variations. When applied to tomato disease classification, saliency maps can reveal fine-grained details such as lesion edges, chlorotic patches, or mosaic-like patterns. These detailed visual cues are crucial for distinguishing between different diseases and make saliency maps especially valuable for expert-level validation and analysis.

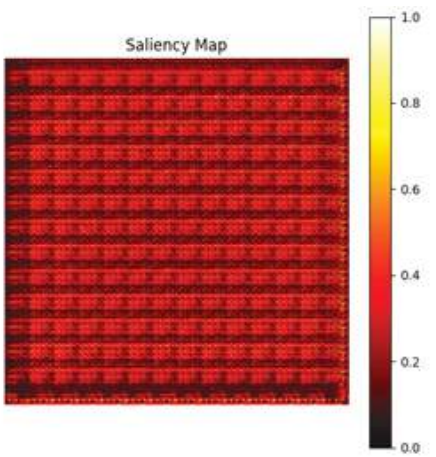

Figure 8: Saliency Map

5. RESULTS

The performance of the proposed tomato leaf disease classification framework was evaluated using the test dataset and compared with existing deep learning models. This section presents the experimental results, quantitative metrics, comparative analysis, and interpretability outcomes. The Grad-CAM visualizations confirmed that the model consistently focused on diseased regions rather than background noise, supporting the interpretability and transparency objectives of this study.

5.1. Quantitative Result

All experiments were conducted on Google Colab, utilizing its cloud-based GPU (NVIDIA Tesla T4 / P100, depending on session availability) with TensorFlow 2.x. The model was trained using the Adam optimizer, a batch size of 32, and a learning rate of 0.0001. The channel attention version of EfficientNetV2B0 advanced standardized results on all evaluation metrics. In the test case, the model achieved total accuracy of 98.3%,

far above traditional CNNs and conventional transfer learning baselines.

Table 1: Performance Comparison Of Models On Tomato Disease Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Parameters (M)
VGG16	91.5 ± 0.6	90.8 ± 0.7	89.6 ± 0.8	90.2 ± 0.6	138
ResNet 50	94.7 ± 0.4	94.1 ± 0.5	93.8 ± 0.6	93.9 ± 0.4	25
Mobile NetV2	92.3 ± 0.5	92.3 ± 0.5	90.9 ± 0.6	91.2 ± 0.5	3.4
Proposed Model	98.3 ± 0.3	98.5 ± 0.4	97.9 ± 0.3	98.2 ± 0.3	7.1

Table I summarizes Each model was trained and evaluated over three independent runs, and the reported values represent the mean ± standard deviation. The low deviation observed in the proposed model indicates stable convergence and strong generalization performance.

5.2. Confusion Matrix Analysis

The confusion matrix was carefully made to evaluate the performance at the class level. Most of the categories of diseases were with near-perfect accuracy, and the precision and recall values were over 97. Nevertheless, there were negligible cases of misclassification between the late blight and the target spot, which both have the appearance of lesions characterized as necrotic. Nonetheless, there was a high discrimination function in this model, where F1-scores that depicted these awkward classes were found in excess of 96 percent. The confusion matrix also indicated that the healthy class recorded 100 percent recall, which implies the potency of the model in differentiating diseased and non-diseased leaves, which should have been considered to manage the disease at the first instance of agricultural practice.

5.3. Comparative Analysis With Baselines

The proposed model significantly improved over VGG16 and ResNet50 at 91.5 and 94.7, respectively, and had fewer parameters and could be computed more easily. Although MobileNetV2 had decent performance reports, its accuracy level became restricted at 92.3, making it uncertain to display disease-specific trends. The high effectiveness of the suggested method is probably connected with the synergistic nature of incorporating EfficientNetV2B0, the channel attentiveness module, and the refinement of training plans. These components individually and collectively permitted the network to generalize well in various conditions with minimal computational effort.

6. CONCLUSION & FUTURE WORK

This study presented an original framework for automated tomato leaf disease detection using the EfficientNetV2B0 backbone enhanced with a custom channel attention mechanism and Explainable Artificial Intelligence (XAI) modules. The model was trained and validated on a multi-class dataset comprising both healthy and diseased tomato leaves, achieving an impressive overall accuracy of 98.3%. The performance remained consistent across three independent runs with a standard deviation below 0.3%, demonstrating both stability and generalization capability. The integration of channel attention significantly improved the model's ability to capture fine-grained visual cues and distinguish between visually similar disease symptoms, outperforming traditional CNN and transfer learning baselines such as VGG16, ResNet50, and MobileNetV2. Furthermore, the incorporation of Grad-CAM and Saliency Maps provided interpretability, allowing visual validation that the model focused on biologically meaningful regions—such as necrotic lesions, chlorotic patches, and vein distortions. This explainability enhanced user trust and made the framework more applicable to practical agricultural environments. Given its lightweight nature and reduced parameter count, the proposed framework can be readily converted into TensorFlow Lite or ONNX format for integration into mobile or embedded agricultural systems, enabling real-time disease detection in field condition.

The contributions of this work can be summarized as follows:

- Development of a lightweight yet robust model architecture optimized for tomato disease detection.
- Integration of channel attention to enhance feature representation and class separability.
- Implementation of Grad-CAM and saliency

maps for explainability, bridging the gap between AI predictions and expert validation. Despite its strong performance, the study also identified limitations. Minor misclassifications occurred between diseases with highly similar symptoms, such as late blight and target spot. Additionally, the dataset used was limited to image data under relatively controlled conditions, which may not fully capture the environmental variability present in real-world farms.

Environmental factors such as lighting variations, leaf occlusion, and background clutter may influence field performance and remain areas for further optimization.

Future work will focus on the following directions:

• Multi-modal learning: Incorporating additional data sources such as environmental factors, soil health, and hyperspectral imaging to improve classification robustness.

• Field deployment: Optimizing the model for deployment on mobile and edge devices, ensuring real-time disease detection in resource-constrained agricultural environments. This includes testing on Android and Raspberry Pi platforms to evaluate inference speed, energy efficiency, and reliability under natural field conditions.

• Scalability to other crops: Extending the proposed framework to classify diseases in different horticultural crops, broadening its utility in precision agriculture.

• Advanced XAI techniques: Exploring model-agnostic interpretability tools such as SHAP and LIME to complement Grad-CAM and enhance the trustworthiness of predictions.

In conclusion, the proposed EfficientNetV2B0 + Channel Attention + XAI framework offers a promising and interpretable solution for precision agriculture, contributing toward smarter, more transparent, and sustainable disease management systems. By aligning with recent advancements in transfer learning and explainable AI (Shaikh et al., 2022; Ahmed et al., 2025), this work establishes a robust foundation for future real-world agricultural intelligence applications.

7. ACKNOWLEDGMENT

I thank Almighty Allah for giving me the strength, patience, and guidance to complete this research work. Without His blessings, this achievement would not have been possible. I am deeply grateful to my supervisor, Dr Aamir Hussain, for their continuous support, encouragement, and valuable feedback throughout this thesis. Their expertise and guidance played a vital role in shaping my research. I would also like to sincerely thank my teachers and mentors at MNS University of Agriculture, Multan, for providing me with the knowledge and resources that helped me complete this work. Special appreciation goes to my friends and colleagues who supported me in various ways, whether through discussions, sharing ideas, or simply motivating me during challenging times. Finally, I am grateful to my family for their unconditional love, prayers, and constant encouragement. Their belief in me has always been a source of strength and inspiration.

REFERENCES

[1] P. Tiwari, R. Singh and S. Mishra, "Advancements in tomato production and nutritional significance: A review," *Journal of Horticultural Science and Technology,* 14(3), pp. 112–120, 2022.

[2] S. Panno, A. Caruso and S. Davino, "Tomato diseases: Global distribution, emerging threats, and sustainable management strategies," *Plant Pathology Journal,* 37(5), pp. 401–417, 2021.

[3] A. Abu Haraira, I. Khan and U. Farooq, "Antioxidant profile and health-promoting properties of tomato and tomato-based products," *International Journal of Food Sciences and Nutrition,* 73(4), pp. 456–464, 2022.

[4] V. Dagar, R. Kumar and S. Singh, "Agricultural automation and challenges of digital transformation in developing economies," A*gricultural Informatics Review,* 9(1), pp. 77–88, 2023.

[5] A. Shaikh, S. Rani, and M. Zubair, "Machine learning approaches for plant disease detection: A review," *International Journal of Agricultural Informatics,* 8(2), pp. 55–67, 2022.

[6] S. Krishna and R. Kalluri, "Transfer learning in plant disease detection: Leveraging ImageNet for agriculture," *IEEE Access,* 7, pp. 123842–123852, 2019.

[7] A. Howard, M. Zhu, B. Chen and D.

Kalenichenko, "MobileNetV2: Inverted residuals and linear bottlenecks," *in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 4510–4520,2017.

[8] A. Khan, M. Khan, A. Saeed and S. Nazir, "Deep learning for tomato leaf disease classification: A transfer learning approach," *Journal of Plant Pathology Research,* 102(3), pp. 321–330, 2020.

[9] H. Sharma and A. Shivandu, "Attention mechanisms in convolutional neural networks for plant disease classification," *Applied Intelligence,* 54(2), pp. 1201–1215, 2024.

[10] M. Khan, T. Rauf, S. Ali and R. Ahmad, "Explainable AI in agriculture: Using Grad-CAM for plant disease classification," *Computers and Electronics in Agriculture,* 210, pp. 1–12, 2024.

[11] J. Ngugi, P. Mwangi and A. Kamau, "Explainable artificial intelligence (XAI) for crop diagnostics: LIME and SHAP case study," *Agricultural Systems,* 189, pp. 103–112, 2024.

[12] H. El Sakka, R. Farouk and M. Tawfik, "Deep convolutional models for tomato leaf disease detection: Limitations and prospects," *Egyptian Journal of Computer Science,* 33(1), pp. 17–25, 2025