Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081477

ARTICLE

Addressing Background Bias in Explainable Orange Fruit Disease Classification Using Deep Learning

Naeem Ullah[1,*], Javed Ali Khan[2], Michelina Ruocco[3], Antonio Della Cioppa[4], Ivanoe De Falco[5] and Giovanna Sannino[5]

[1]Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy
[2]Department of Computer Science, University of Hertfordshire, Hatfield, UK
[3]Institute for Sustainable Plant Protection (IPSP), National Research Council (CNR), Naples, Italy
[4]Natural Computation Lab (NCLab), Department of Information Engineering, Electrical Engineering, and Applied Mathematics (DIEM), University of Salerno, Salerno, Italy
[5]Institute of High Performance Computing and Networking (ICAR), National Research Council (CNR), Naples, Italy

*Corresponding Author: Naeem Ullah. Email: naeem.ullah@unina.it

Received: 03 March 2026; Accepted: 19 May 2026; Published: 15 June 2026

ABSTRACT: Fruit diseases significantly impact agricultural productivity, yet automated detection systems often fail to provide interpretable predictions and are sensitive to background variations in images, particularly in orange fruit disease datasets. Current deep learning approaches are prone to background bias, which reduces explainability and generalization. To address this, we propose a deep learning framework that explicitly reduces background noise and bias in orange fruit disease image classification while providing interpretable, pixel-level predictions. The framework integrates existing architectural components, including grouped convolutions with channel shuffling, Leaky ReLU and clipped ReLU activations, and attention-based feature extraction, within a bias-aware design motivated by explainability analysis. The contribution lies in the problem-driven integration of these components and a background standardization preprocessing step to improve explanation reliability. A Grid Search algorithm is used to optimize the hyperparameters. Data augmentation is applied to enhance generalization. We used perceptual hashing to ensure no duplicate images existed between training and testing sets, thereby preventing data leakage and maintaining dataset integrity. For interpretability, we employ Local Interpretable Model-agnostic Explanations (LIME); however, initial explanations highlighted irrelevant background regions. To address this, we introduce a novel preprocessing step using the GrabCut algorithm and morphological operations to standardize image backgrounds, ensuring explanations focus solely on diseased regions. Unlike existing methods, our background standardization technique, based on GrabCut and white background standardization, improves the relevance of LIME explanations by reducing background-focused attributions from 62.2% to 7.7% of cases, while yielding modest, consistent improvements in classification accuracy (0.15%–0.24%). We further evaluate DeepOrangeNet's feature extraction by classifying its learned representations using six classifiers, including linear discriminant analysis, fine decision tree, Gaussian Naive Bayes, fine k-nearest neighbors, linear support vector machine, and logistic regression, demonstrating its superior adaptability. DeepOrangeNet has been compared with the state-of-the-art methods, proving not only its accuracy but also its explainable and lightweight architecture for real-world agricultural implementation.

KEYWORDS: Background bias; orange fruit disease classification; agricultural productivity; explainable artificial intelligence; deep learning

1 Introduction

The demands for accurate fruit disease detection and classification in the global agriculture sector have increased in recent years [1]. Fruit crop illnesses cause farmers to face severe issues, such as lower yields and hard financial times. As such, identifying diseases and determining their severity is one of the most critical tasks in the agriculture sector [2]. Among the citrus fruits, the orange fruit industry is probably one of the most vital components of the agricultural economy, contributing significantly to both the local and international markets. In fact, the Food and Agriculture Organization (FAO) has estimated that citrus fruit output would reach 197,198 million tons worldwide, and orange fruits, which make up more than half of the overall yield, must be accurately inspected. The quantity and quality of orange fruits are greatly impacted by the broad impacts of several diseases (such as canker, greasy spots, black spots, greening, and many more), which result in substantial economic losses and grave threats to food security [2].

The traditional approaches for fruit disease detection rely heavily on humans, which is labor-intensive, time-consuming, inaccurate, and prone to human error. Automation techniques can help identify infections or diseases in citrus fruit early. This results in better agricultural productivity, cost savings, mitigation of environmental impact, containment and prevention of disease, research and disease management, etc. Researchers around the world have proposed numerous methods for automated fruit disease classification. However, many of these approaches prioritize predictive performance while overlooking interpretability, robustness, and deployment constraints.

The evolution of automated fruit disease detection has transitioned from classical Machine Learning (ML) to Deep Learning (DL), yet significant theoretical and practical limitations persist. Early studies [3–5] primarily relied on traditional image processing and manual feature extraction combined with traditional and advanced ML classifiers. While these methods achieved moderate success, their application boundaries are limited. They often fail under varying lighting conditions and require significant human intervention for feature selection. Furthermore, early approaches relied on traditional image processing and ML techniques, including Support Vector Machine (SVM), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), K-Nearest Neighbors (KNN), and fuzzy logic, achieved moderate success but suffered from limited robustness and generalization [3–6].

With the advent of DL, which has revolutionized the landscape of image classification tasks, there has been further progress also related to citrus fruit disease classification and identification. Recent DL approaches, including ResNet18, GoogleNet, ResNet50, and Long Short-Term Memory (LSTM), etc., have improved classification accuracy [7–11]. However, they often focus primarily on performance while overlooking interpretability and robustness to real-world variations.

Despite the promising progress achieved by recent DL-based approaches, several fundamental limitations remain. Most existing methods are primarily evaluated in controlled experimental settings and primarily focus on maximizing classification accuracy. These approaches often fail to account for robustness to background variability, illumination changes, and real-world acquisition conditions. Moreover, these models are largely treated as black boxes, offering limited insight into the visual cues driving their predictions. As highlighted in recent works, many approaches implicitly learn spurious correlations from background regions rather than disease-specific features, leading to biased predictions and misleading explanations. In addition, some of these approaches suffer from methodological problems such as data leakage.

Specifically, most of the existing works [12,13] trained and tested their models on older versions of publicly available orange disease datasets, with duplicate images across splits, resulting in overlap between training and testing samples. This accidental leakage compromises the validity of the performance evaluation. Furthermore, recent advances in explainable artificial intelligence (XAI) have attempted to improve the

transparency of DL models in agricultural imaging [14]. However, most explanation methods, including post-hoc techniques such as saliency maps and local surrogate models, remain highly sensitive to background artifacts and domain shifts. Domain-adaptive and explanation-aware learning strategies are still underexplored in fruit disease classification, particularly in the context of background-sensitive explanations and deployment in real-field conditions.

To address these challenges, this study proposes DeepOrangeNet, a lightweight deep learning framework designed for accurate and interpretable orange fruit disease classification. Unlike conventional approaches that prioritize accuracy alone, the proposed framework explicitly incorporates explainability into the model development process. In particular, we utilize Local Interpretable Model-agnostic Explanations (LIME) [15] to analyze the decision-making behavior of the model. Initial analysis revealed a strong tendency of the model to focus on irrelevant background regions rather than disease-affected areas, highlighting the presence of background bias.

To mitigate this issue, we introduce a preprocessing strategy that standardizes image backgrounds using the GrabCut algorithm [16] along with morphological operations, ensuring that the model focuses primarily on the fruit region. This preprocessing step is combined with a carefully designed deep learning architecture that balances predictive performance and computational efficiency. The overall framework aims not only to achieve high classification accuracy but also to improve the reliability and interpretability of model explanations, which is essential for real-world agricultural deployment.

The main contributions of this work can be summarized as follows:

- A lightweight deep learning model, DeepOrangeNet, for accurate orange fruit disease classification.
- An explainability-driven analysis that identifies and quantifies background bias in model predictions.
- A preprocessing strategy for background standardization that significantly improves explanation reliability.
- A comprehensive evaluation framework that ensures data integrity and robust performance assessment.

Unlike prior studies, this work emphasizes the importance of integrating explainability with model design and evaluation. By explicitly addressing background-induced bias and improving the alignment between model predictions and biologically relevant features, the proposed approach provides a more transparent and reliable solution for fruit disease classification. The results demonstrate that while accuracy improvements are modest, the gains in explanation quality are substantial, highlighting the practical importance of interpretability in agricultural applications.

The rest of the paper is organized as follows: Section 2 describes the materials and methods used in this study, including the dataset, preprocessing steps, and details of the DeepOrangeNet framework. Section 3 presents the experimental results and performance evaluation. Section 4 discusses the findings in the context of existing research and highlights the contributions and limitations. Finally, Section 5 concludes the paper and outlines potential directions for future work.

2 Materials and Methods

2.1 Dataset

We used the "Orange Diseases Dataset" [17], a freely available dataset that includes images of four classes, including three disease classes (citrus canker, black spot, and greening citrus) and one healthy class, i.e., fresh orange images. Citrus canker is a bacterial disease that affects citrus plants. Symptoms include yellow spots on leaves, deformed fruits, and the formation of lesions or cankers on the bark of trees. The bacterium that causes citrus canker spreads through water, gardening tools, and contact between infected and healthy plants. Citrus canker can significantly reduce the yield and quality of fruits, leading to economic losses for growers.

Citrus black spot is a disease affecting citrus fruits and is caused by a fungus. Infected trees show dark, sunken lesions on the fruit, which can cause premature fruit drop. It can also cause leaf spots and twig dieback. The fungus spreads through spores that can be carried by wind, rain, and contaminated tools. It thrives in warm, humid conditions. Citrus black spot disease can reduce fruit quality and marketability, which causes economic losses for growers. In severe cases, it can affect overall tree health. Citrus greening, also referred to as Huanglongbing (HLB), is a devastating bacterial disease affecting citrus trees. Infected trees exhibit yellowing of leaves, asymmetric fruit coloration, and small, misshapen fruit. Leaves may also show a blotchy pattern. The disease is primarily spread by the Asian citrus psyllid (*Diaphorina citri*); this feeds on the sap of infected trees and transmits the bacterium. Citrus greening causes significant declines in fruit quality and yield and often causes tree death within a few years. It poses a major threat to the citrus industry worldwide.

The dataset comprises 1090 images in PNG and JPG formats. It is divided into test and train folders. The distribution of the images over the classes before augmentations is reported in Table 1, while examples of images contained in the dataset are shown in Fig. 1. It is important to note that the variations in image orientation and rotation observed in Fig. 1 are inherent to the original dataset and were not introduced during preprocessing. These variations reflect real-world image acquisition conditions.

Table 1: Orange diseases dataset image counts before and after training-only augmentation.

Class	Before Augmentation		After Augmentation	
	Training	Testing	Training	Testing
Canker	179	22	1800	22
Greening	347	22	1800	22
Blackspot	184	22	1800	22
Fresh	281	32	1800	32
Total Images	991	98	7200	98

Figure 1: Dataset samples: the first column has an image of a canker, the second shows an image blackspot, the third has a greening image, and the fourth contains a healthy orange.

2.2 Dataset Integrity Validation

To ensure the stability of the experimental setup and eliminate the risk of data leakage, a step is utilized to verify the integrity of the dataset. This process confirms whether overlapping images exist between the training and test sets, something that otherwise may lead to performance metrics that are biased and result in model generalization loss.

We employed a hashing-based image comparison technique with the imagehash library, which computes perceptual hashes (average hash) for each image. The hashes are robust to small variations in image properties and support efficient duplicate detection.

The algorithm continuously computes and stores hash values for all images in the training set and then compares each image in the test set with the hashes. Upon finding a match, the image is flagged as a duplicate, indicating overlap between partitions. This process ensures that no similar or identical images are present in both the training and test subsets.

By preventing duplicates from existing within partitions, we guarantee that model performance is obtained from true generalization and not because of memorization, and thus, we enhance the strength of reported results.

2.3 Data Augmentation

The original dataset exhibits class imbalance, which can bias the model toward majority classes and lead to suboptimal performance on underrepresented classes. To mitigate this issue, we adopt a class-balanced augmentation strategy, ensuring that each class contributes equally during training. This improves class-wise predictive performance and helps reduce overfitting during training. The augmentation process generates multiple variants per image; however, only a subset is retained to reach a fixed target size per class. Any additional augmented samples beyond this limit are not included, as the selected set already provides sufficient diversity. This controlled selection does not negatively impact the final results but instead ensures balanced and stable training across all classes. For the proposed approach, we used a data augmentation approach exclusively on the training set to improve generalization and mitigate class imbalance. No augmentation was applied to the testing set in order to avoid statistical bias and to ensure fair and unbiased performance evaluation.

For this purpose, we employed different data augmentation techniques using the Albumentations library [18] to enhance the number of images, the diversity of the dataset, and the robustness of the model. Data augmentation generates a well-rounded set of image variations, which helps to prevent a model from overfitting. We used rotations, horizontal and vertical flips, shift-scale rotate, and random fog transformations for each image. These transformations are selected to approximate common field-level variations in a label-preserving manner.

In particular, random shifting and scaling introduce partial spatial displacement that reflects changes in camera distance and mild occlusion effects at image boundaries. Whereas fog augmentation and random rotations approximate reduced visibility and sensor orientation variability commonly encountered in outdoor acquisition conditions. All dataset images are randomly rotated within a range of ±30 degrees, producing multiple image variants based on image orientations. To simulate environmental conditions and allow the model to deal with images of different lighting levels, we used random fog with a 20% probability. Furthermore, we used random scaling (up to 10% variation), shifts (up to 5% of the image dimensions), random shifts (up to 5% of the image dimensions), scaling (up to 10% variation), and additional rotations (up to 5 degrees).

Each image in the original dataset is augmented multiple times to reach a target dataset size of 8000 images per class, as shown in Fig. 2. This expansion is intended to balance class distributions and improve training stability, rather than to simulate the full diversity of real-world conditions. While these transformations introduce controlled variations (e.g., orientation, scale, and mild visibility changes), they remain label-preserving and are derived from a limited set of original samples. Therefore, the augmented dataset

should be interpreted as a regularization mechanism to reduce overfitting, rather than a substitute for large-scale, diverse data collection. Additionally, the distribution of the images over the classes after augmentations is reported in Table 1. More complex real-world conditions such as severe leaf occlusion, overlapping fruits, and extreme illumination variations require dedicated data collection or physics-aware augmentation strategies and are therefore left as future work. Consequently, true robustness to real-world variability requires evaluation on larger, independently collected datasets with diverse environmental conditions.

Figure 2: Image augmentation techniques applied to the dataset to improve model generalization. From left to right: original image, rotated, horizontally flipped, vertically flipped, shift-scale-rotate, and random fog.

2.4 Dataset Partitioning

We separated the database into training and testing datasets, as shown in Table 1. We combined a hold-out test set and 5-fold cross-validation (CV) on the remaining data. To test the model on unseen data, we did not use the testing dataset (the hold-out test set) during the training of the proposed model. This offers an unbiased evaluation of the model's performance using never-before-seen data. Then, to train and validate the novel DeepOrangeNet model, we utilize the training data in a 5-fold (K-fold) CV setting, where the training will be repeated five times. Because of the use of a 5-fold CV, the database will be separated into five equal collections, and the testing set will be different every time. It offers a more precise assessment of how well our model is expected to perform in real-world circumstances and helps prevent problems like overfitting. However, the current implementation uses non-stratified fold splits, which may introduce slight variations in class distributions across folds. While all models are evaluated using identical splits to ensure fair comparison, this may affect the stability of validation metrics. Future work will adopt stratified cross-validation to further improve evaluation reliability.

2.5 Proposed DeepOrangeNet Approach

DeepOrangeNet is an explainability-guided and bias-aware DL framework designed for reliable classification of orange diseases. We propose a hybrid approach that combines DL, grid search, and eXplainable Artificial Intelligence (XAI) as Local Interpretable Model-agnostic Explanations (LIME) [15] techniques to improve optimization, performance, and transparency of the orange disease classification task. Fig. 3 provides an overview of the proposed approach.

2.6 Deep Architecture Details

The architecture of the proposed DeepOrangeNet model resulted from the iterative process of insight from the literature, trial and error, and systematic evaluation provided by an ablation study. The architectural design of DeepOrangeNet was guided primarily by the need to reduce background-dependent predictions revealed during explainability analysis. Lightweight operations such as grouped convolutions and channel shuffling were therefore adopted as engineering choices to maintain computational efficiency, rather than introduced as novel mechanisms.

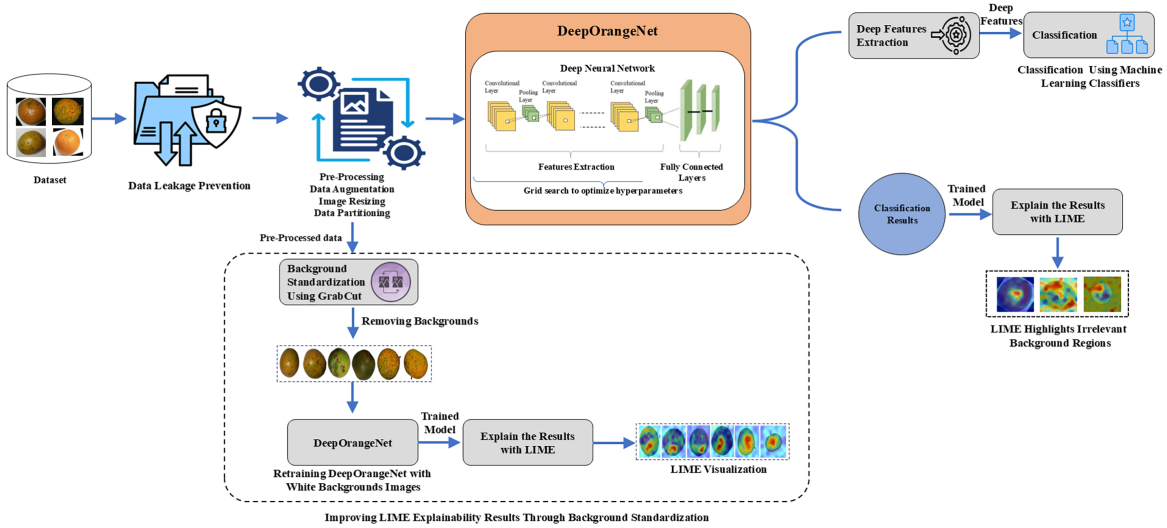

Figure 3: Workflow of the proposed approach.

Additionally, we performed trial and error in combining convolutional layers, activation functions, and pooling strategies to analyze their impact on feature extraction. Such an approach helps identify and avoid redundant components, such as too many fully connected layers, resulting in overfitting and negligible performance improvement. We used ShuffleNet units to manage model complexity and support deployment in resource-constrained agricultural environments.

The Fire module adopts a squeeze-expand design in which a 1 × 1 convolution reduces channel dimensionality (squeeze), followed by a combination of 1 × 1 and 3 × 3 convolutions (expand). In this work, a squeeze-to-expand ratio of 1:4 is used (e.g., 32 squeeze filters and 128 expand filters). This ratio is commonly used in lightweight architectures to balance parameter efficiency and representational capacity. The selected configuration was empirically validated during preliminary experiments, where it provided stable performance without increasing computational complexity.

Furthermore, we performed an ablation study after experimentation to refine the architecture, systematically investigating the influence of each component on the classification performance. We analyzed the variations in the number of convolutional and fully connected layers and the inclusion of ShuffleNet units and fire modules. During this ablation process, architectural parameters such as the number of groups in grouped convolutions and the self-attention configuration (multi-head attention with 8 heads and embedding dimension of 64) were empirically fixed based on stability and performance trade-offs, rather than treated as tunable hyperparameters. From these validations, we complete the architecture with 20 learned layers, including 17 Conv layers, two shufflenet units (including three Conv and one channel shuffling layer), and 1 FC layer. The structure of the DeepOrangeNet architecture is shown in Table 2.

Table 2: DeepOrangeNet architecture details.

S. No.	Operation	Layers	Filter Size	No. of Filters	Padding	Stride
1		Input	Input Layer			
2	Convolution	Convolution (LReLU, CCN)	11 × 11	96		4 × 4

(Continued)

Table 2 (continued)

S. No.	Operation	Layers	Filter Size	No. of Filters	Padding	Stride
3	Pooling	Max pooling	3 × 3			2 × 2
4	Convolution	Group convolution (LReLU, CCN)	5 × 5	128	2 × 2	
5	Pooling	Max pooling	3 × 3			2 × 2
6	Convolution	Group convolution (LReLU, BN)	3 × 3	64	1 × 1	
7	Convolution	Group convolution (LReLU, BN)	3 × 3	64	1 × 1	
8	Convolution	Group convolution (LReLU, BN)	3 × 3	384	1 × 1	
9	Convolution	Group convolution (LReLU, BN)	3 × 3	192	1 × 1	
10	Shufflenet unit	Convolution (BN)	1 × 1	32		
		Channel shuffling layer				
		Convolution (BN)	3 × 3	128	1 × 1	
		Convolution (BN)	1 × 1	128		
11	Pooling	Max pooling	3 × 3			2 × 2
12	Shufflenet unit	Convolution (BN)	1 × 1	68		
		Channel shuffling layer				
		Convolution (BN)	3 × 3	1	1 × 1	
		Convolution (BN)	1 × 1	68		
13	Convolution	Group convolution (LReLU, CCN)	3 × 3	128	1 × 1	
14	Convolution	Convolution (Clipped LReLU, BN)	3 × 3	690		
15	Group Convolution	Convolution (Clipped LReLU, BN)	3 × 3	1		
16	Convolution	Convolution (BN)	1 × 1	320		
17	Convolution	Convolution (Clipped LReLU, BN)	1 × 1	1280		
18		Flatten Layer				

(Continued)

Table 2 (continued)

S. No.	Operation	Layers	Filter Size	No. of Filters	Padding	Stride
19		SelfAttention Layer				
20		FC + Softmax + Classification				

To further clarify the architectural configuration presented in Table 2, we provide additional details regarding the parameter selection of key components. The grouped convolution layers employ varying group sizes (g = 2 and g = 4) depending on the network stage. Smaller group sizes are used in earlier layers to preserve feature diversity, while larger groups are introduced in deeper layers to improve computational efficiency. In certain deeper layers, depthwise-style grouping is adopted to further reduce complexity.

The ShuffleNet units follow a lightweight bottleneck structure consisting of pointwise grouped convolution (1 × 1), channel shuffling, and depthwise convolution (3 × 3). Channel shuffling is performed with 4 groups to enable effective inter-group information exchange. The filter sizes (e.g., 32, 64, 128) are progressively increased across layers to enhance feature representation capacity while maintaining efficiency. These parameters were empirically selected based on stability and performance trade-offs observed during preliminary experiments.

The initial layer contains I × J units, where *I* and *J* refer to the height and width of the input image, respectively. This layer matches the actual size of the input images, which are of dimension 227 × 227 pixels. Conv layers with filter sizes of 5 × 5, 11 × 11, 3 × 3, and 1 × 1 are applied to the input images to create the feature maps. Standard convolution operations are applied to extract feature maps from input images of size 227 × 227 pixels.

We used Batch Normalization (BN) and Cross Channel Normalization (CCN) to regularize the inputs, offer regularization, and improve generalization ability. We used Leaky ReLU (LReLU) to allow a small non-zero gradient for negative inputs (f(x) = max(0.01x, x)), preventing dying neuron issues common with standard ReLU. We also used clipped ReLU to bound activations and prevent gradient explosion. Max-pooling layers (2 × 2 stride) are used for down-sampling.

The feature extraction module, which comprises Conv and GConv layers, is followed by flattened and self-attention layers.

The self-attention module is implemented as a multi-head self-attention mechanism applied after feature flattening. Specifically, the attention layer uses 8 attention heads with an embedding dimension of 64. This configuration enables the model to capture global dependencies between feature channels and enhances its ability to focus on disease-relevant regions while suppressing background noise. The use of multi-head attention allows the model to learn diverse feature interactions with minimal computational overhead. The fully connected layer transforms the flattened feature map into a 1D feature vector. The softmax and classification layers follow the FC layer. The layer output is sent to a 4-way softmax because our dataset has four categories.

2.7 Hyperparameters Optimization

The effectiveness of DL frameworks depends on the selection of appropriate hyperparameters. In this study, hyperparameter optimization was conducted using a grid search strategy focused on learning rate and training epochs, as these parameters have the most direct influence on convergence behavior and training stability. Specifically, we used the learning rates of 0.01, 0.1, and 0.001 and training epochs of 20, 30, 40, 45,

and 50. However, other architectural parameters, such as the number of groups in grouped convolutions and the configuration of the self-attention module, were treated as fixed design choices determined through prior architectural exploration and ablation analysis (Section 2.6). Fixing these parameters reduces the risk of overfitting, limits the search space, and ensures architectural consistency across cross-validation folds. To control computational cost, grid search was performed before cross-validation, and the selected hyperparameters were subsequently applied uniformly across all folds. The hyperparameter values reported in Table 3 were selected through a combination of experimental tuning and established practices. In particular, the learning rate and number of epochs were determined using grid search based on validation performance. Other parameters, such as the choice of optimizer (SGDM), dropout rate, and activation function (LReLU), were selected based on commonly adopted settings in prior deep learning studies. This hybrid strategy ensures both empirical effectiveness and consistency with established methodologies. The final hyperparameter configuration is reported in Table 3. A learning rate of 0.001 was selected based on grid search experiments conducted on a subset of the dataset. This value provided stable and consistent optimization performance across training runs. Model convergence and training stability are further assessed using training and validation accuracy and loss trends.

Table 3: Hyperparameters of the proposed architecture.

Parameters	Values
AF	LReLU
Shuffle	Every epoch
Maximum Epochs	45
Optimization algorithm	SGDM
K-fold	5
Dropout	0.5
Iterations per epoch	12
Learning rate	0.001
Validation frequency	30

2.8 Explainability

Though the DL's inherent complexity can occasionally make it challenging to understand why and how certain predictions are generated, these systems can demonstrate outstanding predictive power in the interpretation of pictures. Black-box CNN methodologies have long been criticized for being difficult to understand despite their effectiveness. Because of their opacity, these models may not be adopted in therapeutic settings since it is difficult to fully comprehend and believe in their decisions. In this instance, XAI is the crucial response. XAI seeks to improve the transparency of complex AI models' decision-making procedures.

For a model to be considered explainable, it must be clear which parts of the input data (images of orange fruit) the model uses for decision-making and how much these parts contribute to the output. We analyzed three state-of-the-art explainability approaches: LIME, Gradient-weighted Class Activation Mapping (Grad-CAM) [19], and Gradient Attribution [20].

While Grad-CAM and Gradient Attribution were implemented, they produced coarse saliency maps that lacked the spatial precision necessary to identify small-scale pathological features, such as specific canker lesions or tiny black spots. Grad-CAM heatmaps often highlighted broad regions of the fruit, sometimes including healthy tissue or background, due to the low resolution of the final convolutional

layers. In contrast, LIME uses a superpixel-based perturbation approach, which allows for more granular, feature-level interpretations. This was found to be more relevant for agricultural experts, as it clearly distinguishes diseased regions from the surrounding healthy rind. Furthermore, LIME substantially reduces background-focused explanations after preprocessing, confirming its ability to provide precise and interpretable region-level explanations.

Consequently, LIME was selected as the primary XAI tool to ensure that the model's regions of interest (ROI) align closely with established diagnostic markers. More advanced CAM-based variants, such as Grad-CAM++ and Score-CAM, were not included in the current study because our primary objective was to analyze background bias and region-level attribution consistency, rather than to benchmark all available XAI methods.

Local Interpretable Model-Agnostic Explanations (LIME)

LIME is primarily concerned with producing locally accurate explanations for individual predictions through input data perturbation and output model observation. It provides insights into feature relevance by constructing a local surrogate model to imitate the intricate behavior of the underlying model. LIME's benefit is that it might provide comprehensible, localized explanations for predictions made by complex ML and DL models, such as black-box CNNs. We might gain insights into the DL model's decision-making process by using LIME to focus on the regions of interest in orange fruit that influenced the model's predictions. This tactic makes the model more transparent and provides agricultural experts and users with a comprehensive understanding of its operation. The advantages of XAI, such as increased model reliability and the ability to analyze complex models, further emphasize the necessity of using such technology.

3 Results

To evaluate the proposed DeepOrangeNet model's effectiveness, we conducted multiple tests on the "Orange diseases dataset", a publicly available dataset from Kaggle. First, to carry out the experiments, we used an image scaling method and resized images into 227 by 227 pixels to increase processing speed and comply with the framework's input layer parameters.

As evaluation metrics, we employed accuracy, precision, sensitivity, and F1-score measures, and all experiments were performed on a computer whose properties are reported in Table 4. On this specific computer, the DeepOrangeNet framework has requested 562 min and 51 s to train. However, this time depends on the number of folds (k-fold, i.e., repetitions) epochs and iterations and on the computing power utilized. The DeepOrangeNet underwent 450 iterations (in each repetition) during the training stage—ten iterations each epoch—for 45 epochs in one round of five-fold cross-validation.

Table 4: Details of the system used for the model implementation and for running the tests.

Name	Hardware/Software Configurations
HDD	500 GB
RAM	8 GB
SSD	232 GB
Type of System	64-bit, Windows 10
Development tool	MATLAB R2020a
CPU	Intel (R) Core (TM) i5-5200U

3.1 Results on the Original Dataset

The purpose of this experiment is to establish a baseline performance with limited data (original dataset) and evaluate the performance of the proposed DeepOrangeNet model without employing any data augmentation techniques. Table 5 has five-fold cross-validation classification results on the original dataset, whereas Table 6 has testing results on unseen samples, demonstrating a strong performance across all the evaluated metrics. The model achieved an average accuracy of 97.55% during cross-validation (Table 5), with relatively low variance (0.11), indicating consistent performance across folds. However, in the case of unseen samples, the proposed DeepOrangeNet approach achieved an average accuracy of 97.38%. Overall, these results indicate that the model performs well on the original dataset, but the limited amount of training data may restrict its ability to achieve higher performance. This experiment highlighted the need for increased training data, leading to the second experiment involving a data-augmented version of the dataset to address these limitations.

Table 5: Five-fold cross-validation results on the original dataset.

Metric	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Average	97.55	94.9	95.30	94.80
Median	97.47	95.00	95.25	94.25
Maximum	97.98	95.50	95.75	95.75
Minimum	97.08	94.25	94.75	94.25
Standard deviation	0.33	0.62	0.37	0.76
Variance	0.11	0.39	0.14	0.56

Table 6: Testing on unseen samples results on the original dataset.

Metric	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Average	97.38	93.05	93.30	93.00
Median	97.47	93.25	93.5	93.25
Maximum	98.23	95.25	95.25	95.25
Minimum	96.23	89.75	90.00	89.75
Standard deviation	0.80	2.19	2.01	2.15
Variance	0.64	4.82	4.04	4.63

3.2 Results on Augmented Dataset

In this experiment, the data augmentation strategy is utilized to broaden the dataset's variety and resolve the limited dataset issue. We trained the model on 7200 augmented images and tested it on 98 augmented images. The five-fold CV results on the augmented dataset are shown in Table 7. We achieved an impressive average validation accuracy of 99.45% and variance of 0.01 which validate a high level of consistency across the validation folds. Furthermore, low standard deviations of 0.22 for recall, precision, and F1-score show consistent predictive performance across folds. These results suggest that data augmentation has enhanced the model's ability to generalize on the validation set effectively.

The results of testing on unseen samples are shown in Table 8, and the model achieved an average accuracy of 98.21%. This accuracy is still strong, even if it is somewhat below the validation set findings. Stable results over unseen samples are shown by the consistently high average accuracy, recall, and F1-score,

as well as the median values that nearly match the corresponding averages. Although it is marginally more on the validation set, the performance variation on unknown data is still within a reasonable range. Even after augmentation, this consistency demonstrates how well the model applies its acquired characteristics to fresh data.

Table 7: Five-Fold cross-validation results on an augmented dataset (validation).

Metric	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Average	99.45	98.90	98.90	98.90
Median	99.38	98.75	98.75	98.75
Maximum	99.62	99.25	99.25	99.25
Minimum	99.37	98.75	98.75	98.75
Standard deviation	0.11	0.22	0.22	0.22
Variance	0.01	0.05	0.05	0.05

Table 8: Testing on unseen samples results on the augmented dataset.

Metric	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Average	98.21	96.50	96.50	96.50
Median	98.12	96.25	96.25	96.25
Maximum	99.07	98.00	98.25	98.00
Minimum	97.69	95.50	95.50	95.50
Standard deviation	0.53	0.95	1.04	0.94
Variance	0.28	0.91	1.09	0.89

From the above findings, it can be concluded that the proposed DeepOrangeNet approach achieved performance improvements across all important measures on the augmented dataset when compared to the baseline findings on the original dataset. The model's average accuracy improved by 1.90% for five-fold cross-validation, from 97.55% for the original dataset to 99.45% for the augmented dataset. With decreases in variance, precision, recall, and F1-score also experienced significant gains of almost 10% on average, demonstrating improved stability and generalization. Furthermore, the average testing (unseen samples) accuracy on the augmented dataset showed an improvement of 0.83%, the accuracy of the model on the enhanced dataset was 98.21%, whereas the accuracy on the original dataset was 97.38%. The performance improvements on the augmented dataset suggest that the augmentations strengthened the model's resistance to changes in unseen data. The consistent improvements in model classification performance demonstrate how beneficial data augmentation is for enhancing the model's classification performance.

3.3 Ablation Study

The primary motivation of this ablation study is to systematically analyze the contribution of each architectural component of DeepOrangeNet and to justify the design choices of the proposed model. By progressively modifying or removing specific components, we aim to understand their individual and combined impact on classification performance [21]. This analysis not only validates the effectiveness of the proposed architecture but also demonstrates how different components, including convolutional layers, FC layers, Fire modules, and ShuffleNet units, contribute to accuracy, feature representation, and

computational efficiency. Four controlled architectural variants are evaluated to examine the contribution of key network components and their interactions. The results of the ablated frameworks on the testing dataset are summarized in Table 9. The outcomes demonstrate that altering or eliminating any part of the DeepOrangeNet model makes the framework perform poorly.

Table 9: Ablation study results on the testing dataset achieved by our DeepOrangeNet model. The input image size is 227 × 227 for all experiments, while the activation function is always LReLU and Clipped ReLU.

Exp. No.	No. of Conv Layers	No. of FC Layers	ShuffleNet Units	Fire Modules	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	13	1	2	0	98.06	96.25	96.25	96.25
2	10	1	1	0	97.50	96.00	96.00	96.00
3	17	4	2	1	95.43	90.80	91.05	90.80
4	17	1	2	1	98.45	96.86	96.70	96.77

Experiment 4, which is the configuration of the proposed model and consists of 17 convolutional layers, 1 FC, 2 ShuffleNet units, and 1 Fire module, achieves the highest overall accuracy of 98.21%, along with precision, recall, and F1-score values of 96.50% each. Whereas the experiment using 13 convolutional layers, 1 FC layer, 2 ShuffleNet units, and no Fire module achieves the second highest accuracy of 98.06%. However, it performs somewhat worse in terms of precision, recall, and F1 score. Experiments 2 and 3 show lower accuracy and performance metrics because they use fewer ShuffleNet units or have more FC layers. The results confirm that combining 17 convolutional layers with a single Fire module and two ShuffleNet units, as described in Experiment 4, yields the best classification performance.

Although Table 9 reports the performance impact of architectural variations, we clarify that the primary role of the Fire module in DeepOrangeNet is architectural efficiency rather than only increasing network depth. The Fire module employs a squeeze-expand strategy that reduces parameter dimensionality through 1 × 1 convolutions before selectively expanding feature representations using a mix of 1 × 1 and 3 × 3 filters. This design encourages compact feature reuse while preserving discriminative capacity, which is particularly beneficial for fine-grained citrus disease patterns. However, to isolate its operational effect, comparison between Experiments 1 and 4 is particularly informative, as both configurations use identical numbers of ShuffleNet units and fully connected layers. The observed performance gain in Experiment 4, therefore, reflects the Fire module's ability to enhance feature abstraction and channel-wise interaction without substantially increasing computational cost. Furthermore, we acknowledge that a fully orthogonal ablation, where the Fire module is toggled while all other architectural components remain fixed, would provide an even clearer quantification of its isolated contribution and will be considered in future work.

To further explore the impact of each component individually, we extend the ablation study with a single-variable control experiment. Although the above experiments tested the combined effects of the architectural changes, they do not individually assess each component's impact. Thus, starting with the complete configuration of the DeepOrangeNet architecture (Experiment B), we individually remove each component, i.e., the grouped convolution, channel shuffling, attention mechanism, and Fire module, while keeping all other settings fixed. The results are presented in Table 10.

Table 10: Single-variable ablation study of DeepOrangeNet showing the contribution of each core component.

Exp	Grouped Conv	Channel Shuffle	Attention	Fire Module	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
B (Full)	✓	✓	✓	✓	98.45	96.86	96.70	96.77
E1	×	✓	✓	✓	96.90	96.00	96.00	96.00
E2	✓	×	✓	✓	97.00	95.00	95.00	95.00
E3	✓	✓	×	✓	96.90	96.00	96.00	96.00
E4	✓	✓	✓	×	97.50	95.50	95.50	95.50

The experiments demonstrate the importance of each component in the overall performance of the model. For instance, the removal of grouped convolutions (E1) leads to a significant decrease in accuracy, which confirms their role in the efficient extraction of features. Similarly, the removal of channel shuffling (E2) leads to a decrease in performance, which confirms the importance of this component in the efficient flow of information between groups. Moreover, the removal of the attention mechanism (E3) leads to a decrease in performance, which confirms the importance of this component in the focus on disease-related areas. Finally, the removal of the Fire module (E4) leads to a decrease in accuracy, which confirms the importance of this component in the efficient representation of features. Overall, the proposed model (B) with all components achieves the best performance, which confirms the importance of integrating all components in the proposed DeepOrangeNet model.

3.4 DeepOrangeNet Feature Evaluation with Traditional Classifiers

In this experiment, we aim to evaluate the feature extraction capability of the DeepOrangeNet model and the effectiveness of the extracted features. We aim to assess whether the proposed DeepOrangeNet model can extract more informative, discriminative, and robust features to achieve satisfactory performance, even when combined with traditional ML classifiers. By comparing DeepOrangeNet with simpler, faster, and more interpretable ML classifiers, which require less computational power than deep learning models, this experiment highlights the trade-offs between overall accuracy and class-specific metrics (precision, recall, and F1-score). This comparison offers insights into the practical benefits of using ML classifiers for quick and cost-effective classification, even on less powerful systems.

In this experiment, we used the proposed DeepOrangeNet model for feature extraction and multiple classification models to detect orange fruit disease. We extracted the features from the first fully connected layer of the DeepOrangeNet model (trained on the augmented dataset) in the form of a feature vector having 1096 features. The number of features (1096) corresponds to the number of units in the first fully connected layer of the DeepOrangeNet model, which is determined by the architecture of the network and the dimensions of the input data. The selected features are then input into six classifiers including fine tree, gaussian naïve Bayes [22], fine KNN [23], Efficient linear SVM [24], Linear Discriminant [25], and Efficient logistic regression [26]. According to the results of this experiment (Table 11), the proposed DeepOrangeNet model achieved the highest testing accuracy of 98.21% whereas Gaussian Naive Bayes and Efficient logistic regression achieved the second highest accuracy of 97.76%.

Furthermore, the proposed DeepOrangeNet model achieved lower precision, recall, and F1-score by 1.25% compared to Gaussian Naive Bayes, which could be attributed to the different ways these models handle features for making predictions. Gaussian Naive Bayes performs well with the extracted deep features (using DeepOrangeNet) because it assumes feature independence, which might be well-aligned with the feature

distribution in this dataset. In contrast, DeepOrangeNet is optimized primarily for overall accuracy, which might not prioritize class-specific metrics as much as traditional ML classifiers.

Table 11: DeepOrangeNet classification performance with traditional ML classifiers using unseen samples.

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Fine tree	97.01	95.25	95.25	95.25
Gaussian naive bayes	97.76	97.75	97.75	97.75
Fine KNN	97.65	97.0	97.0	97.0
Efficient linear SVM	97.65	97.0	97.0	97.0
Linear Discriminant	97.31	95.5	95.5	95.5
Efficient logistic regression	97.76	96.75	96.75	96.75
DeepOrangeNet	98.21	96.50	96.50	96.50

This can also partly be explained due to class imbalance and decision boundary behavior. DL models like DeepOrangeNet are more confident in their outputs due to the softmax activation on the output layer, and this can lead to overconfident misclassifications of underrepresented classes. On the other hand, Gaussian Naive Bayes, in estimating class-conditional probabilities and in assuming feature independence, might be able to handle minority classes more modestly, yielding slightly higher precision and recall. Second, it may also be due to the feature vectors in the deep feature space not being well-separated. They may not form perfectly separable clusters for all classes, leading to suboptimal boundary decisions in end-to-end deep models for certain classes, such that DeepOrangeNet is less discriminatory regarding those classes compared to GNB's probabilistic handling.

Although Gaussian Naive Bayes outperformed DeepOrangeNet by a slight margin on precision, recall, and F1 score, DeepOrangeNet achieved a higher overall accuracy of 0.45%, indicating that it better captures the general trend across all samples. The choice between these methods ultimately depends on the application's priorities—whether the focus is on maximizing overall accuracy or optimizing for more balanced class performance, which can be critical when considering imbalanced datasets or specific metrics like precision, recall, and F1-score. Therefore, in cases where achieving a balance between class-specific metrics is crucial, Gaussian Naive Bayes may be more suitable, despite its slightly lower overall accuracy compared to DeepOrangeNet.

3.5 Explainability Results

Although the DeepOrangeNet model performs well in orange disease classification, it is difficult to understand the logic behind a network's particular output. To reduce this explainability gap and raise the proposed model's dependability, we used LIME method and created gradient maps that resemble Gradient-weighted Class Activation Mapping (Grad-CAM) using the jet color scheme, as shown in Fig. 4. The varying hues in these images correspond to varying levels of importance within the image of the orange fruit. Solid colors like red highlight important areas, and cooler tones like blue represent regions of lesser value and serve as the framework's guidelines for making predictions. Agricultural specialists may identify important disease regions that are critical to the model's classifications by using these maps, which provide a visual breakdown of the model's emphasis.

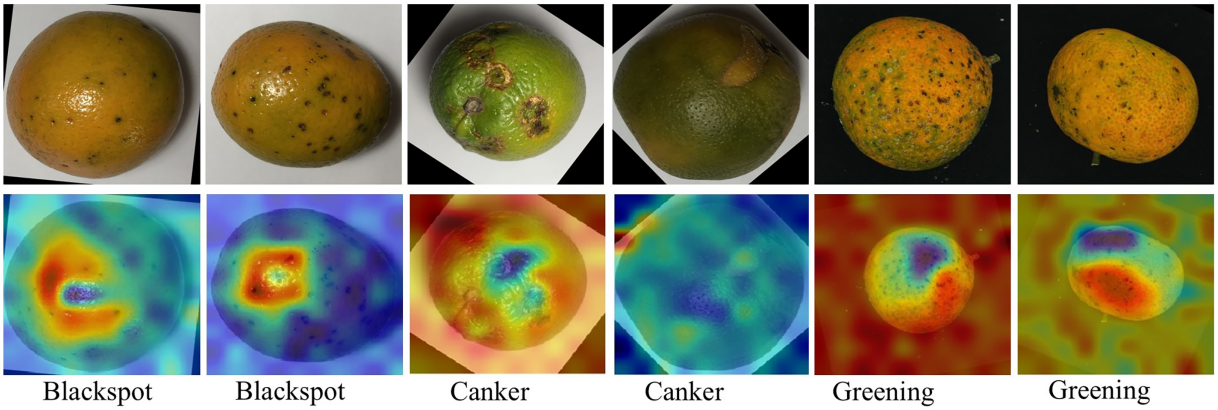

Figure 4: Dataset samples and LIME results using DeepOrangeNet model: the first two columns have black spots, the third and fourth columns have canker images, and the fifth and sixth columns have greening images.

Moreover, beyond region-level highlighting, the LIME explanations can be related to well-known visual disease characteristics in specific ways. For instance, for citrus canker, the highlighted high-importance regions typically correspond to raised, irregular lesion boundaries and surrounding chlorotic tissue. In the case of citrus black spot, the highlighted areas may correspond to sunken, dark necrotic spots on the fruit surface. Whereas, for citrus greening, it may correspond to uneven patterns of discoloration and blotchy pigmentation. Although LIME does not explicitly label biological traits, the regions it highlights are consistent with known pathological manifestations used by human experts for disease identification.

However, from the explainability results, we noticed a problem that seems to show that the proposed DeepOrangeNet model focuses on the background areas rather than on the fruit itself. Highlighting highly the background areas, as done in columns 3 and 5 in Fig. 4, provides one with some concerns about the model using non-relevant cues to classify. Thus, this observation shows that the model might be cheating by using patterns or cues in the background to classify the images, even though these cues are not meaningful or reliable in real-world scenarios. This could lead to problems when the model is exposed to new images with different or varied backgrounds.

To solve this problem and make the model more explainable, in the next section, we present a standardization method for the background. The approach tries to make the model less sensitive to background variations, allowing it to pay attention to the important parts only—for instance, the orange fruit. The challenge being discussed here lays the ground for discussing, in the next subsection, how background standardization improves the model interpretability and its performance overall.

3.5.1 Background Standardization via GrabCut

This experiment aims to overcome the limitation of LIME (instead of highlighting orange regions, LIME highlights unimportant background regions). This indicates that LIME overlooked the orange characteristics in the images and identified background regions as relevant for orange image classification. To do this, we preprocessed our dataset so that every image's background was uniformly white. This phase aims to improve the proposed DeepOrangeNet approach's explainability and performance by eliminating background variability. It is believed that in this instance, the model will predominantly pay attention to the orange portion of the images—that is, their features.

To ensure a consistent white background across the dataset, minimize distractions, and guarantee that the proposed DeepOrangeNet model only focuses on the orange area, we utilized a preprocessing

approach that employs the GrabCut algorithm [16] in OpenCV [27]. The process starts by loading each image and defining a bounding rectangle to roughly encompass the object of interest (the orange). GrabCut is then applied to distinguish the foreground (orange) from the background, followed by morphological operations like closing and dilation using an elliptical kernel to refine the mask and remove noise. Contour detection is used to identify the largest object in the image, assumed to be the orange, and the mask is further refined to focus on this object. The refined mask is then applied to extract the foreground, which is combined with a white background using NumPy operations. The final processed image is saved with a consistent white background. This method ensures accurate object isolation and standardizes the dataset's visual characteristics.

To evaluate the reliability of the GrabCut-based preprocessing, we manually inspected all 1090 images after segmentation. The results showed that approximately 7.12% of the images contained segmentation imperfections. Within this subset, most cases involved small leftover parts of the original background or slightly inaccurate object boundaries. However, a smaller portion (approximately 1.35% of the total images, included within the 7.12%) exhibited more noticeable errors, where larger portions of the background were not fully removed. These errors were typically located around the outer regions of the fruit and rarely overlapped significantly with the central disease-affected areas. The remaining 92.88% of the images were successfully segmented with clean foreground extraction. Overall, this indicates that the preprocessing pipeline is largely robust, although a small proportion of cases may include noticeable background artifacts. Importantly, these imperfections did not significantly affect classification performance or alter the overall trend of improved explainability observed after background standardization.

The preprocessed dataset with uniform white backgrounds is then used to retrain the proposed model. The impact of this transformation is evaluated both quantitatively and qualitatively in the following subsections.

3.5.2 Quantitative Performance Evaluation after Background Standardization

We retrained the DeepOrangeNet model using the preprocessed augmented dataset with a uniform white background. A thorough assessment of the model's performance both before and after preprocessing using five-fold cross-validation is performed in this experiment. The training parameters remained consistent with the original setup shown in Table 3. Table 12, ensuring a fair comparison of results before and after preprocessing. The results in Table 12 summarize the validation and testing results on the augmented dataset with uniform backgrounds, highlighting the impact of converting image backgrounds to white on model performance. These results indicate that background standardization leads to consistent, though numerically modest, improvements in predictive performance while substantially enhancing model robustness and explanation reliability. Accuracy for validation increased from 99.45% to 99.60% (0.15% improvement), while testing accuracy increased from 98.21% to 98.45% (0.24% improvement). Precision resulted in a 0.36% relative improvement and thus can be more assertive in identifying true positives while minimizing false positives. Recall and F1-score showed relative increases of 0.20% and 0.27%, respectively, indicating a better balance between sensitivity and precision.

While the observed accuracy gains are modest in magnitude (0.15% on validation and 0.24% on testing), they are consistent across all evaluated metrics and cross-validation folds. As we have not conducted formal statistical significance testing, we are not considering these numerical differences as strong statistical evidence of superior predictive performance. The primary practical value of background standardization lies in its effect on model interpretability rather than accuracy alone as discussed in detail in Section 3.5.3.

Table 12: Comparison of the validation and testing results before and after dataset standardization.

Metric	Before Background Standardization (Validation)	After Background Standardization (Validation)	Before Background Standardization (Testing)	After Background Standardization (Testing)
Accuracy(%)	99.45	99.60	98.21	98.45
Precision(%)	98.90	99.15	96.50	96.86
Recall(%)	98.90	98.95	96.50	96.70
F1-score(%)	98.90	99.04	96.50	96.77

These results demonstrate that standardizing the dataset by converting backgrounds to white led to consistent, albeit modest, improvements in all performance metrics, indicating its effectiveness in enhancing model predictions.

To further examine the optimization behavior of the model after background standardization, the training and validation accuracy and loss curves are illustrated in Fig. 5. The curves show a smooth and stable convergence pattern, with training and validation accuracy steadily increasing while the corresponding losses decrease across epochs. The absence of significant divergence between training and validation trends indicates that the model maintains good generalization and does not suffer from overfitting. These observations confirm that the selected fixed learning rate and training configuration enable effective optimization of the model.

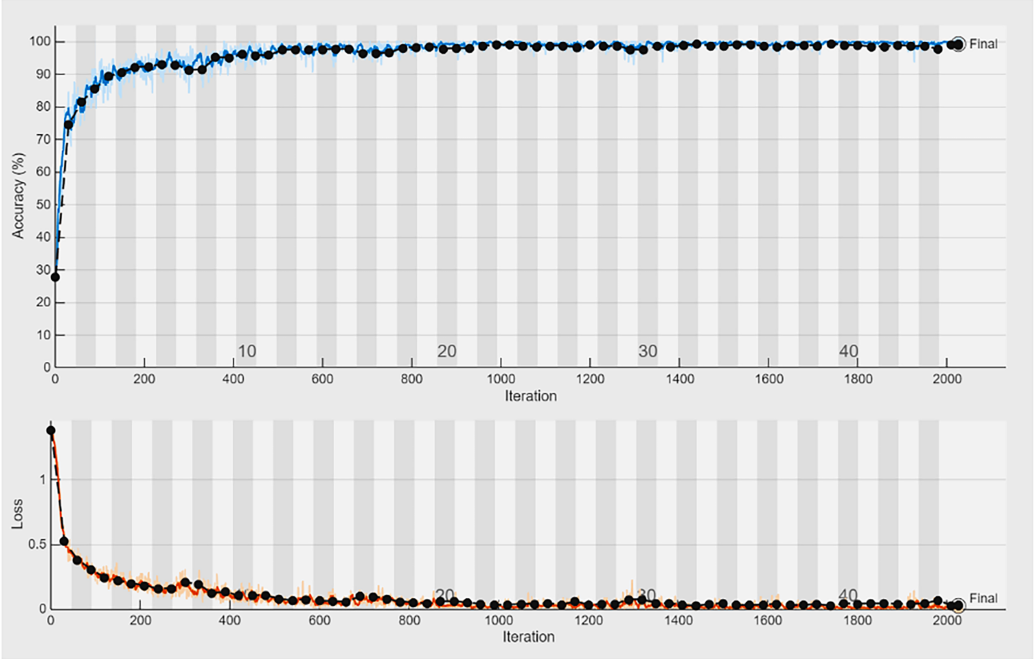

Figure 5: Training and validation accuracy and loss curves of DeepOrangeNet over 45 epochs after background standardization, demonstrating stable convergence and effective optimization behavior. In the accuracy plot (**top**), the solid blue line represents the smoothed training accuracy, the light blue line represents the raw training accuracy, and the black dashed line with circular markers represents the validation accuracy. In the loss plot (**bottom**), the solid red line represents the smoothed training loss, the light orange line represents the raw training loss, and the black dashed line with circular markers represents the validation loss.

To provide a clearer class-wise evaluation and to explicitly demonstrate inter-class confusion behavior, Table 13 presents the aggregated confusion matrix obtained from five-fold cross-validation using the final ensemble model with majority voting. It is important to clarify that the dataset contains four classes only: Blackspot, Canker, Fresh (disease-free), and Greening. The matrix shows that the model achieves near-perfect discrimination across all categories, with only two misclassifications occurring between Blackspot and Canker. No confusion is observed between Fresh samples and diseased categories, nor between Greening and other disease types. These results confirm that the proposed DeepOrangeNet architecture does not merely achieve high overall accuracy but also maintains strong class-level separability, demonstrating meaningful differentiation between visually similar disease patterns.

Table 13: Aggregated confusion matrix (Five-Fold cross-validation) for the final ensemble model (Majority voting). Rows represent true classes and columns represent predicted classes.

True Class	Blackspot	Canker	Fresh	Greening
Blackspot	20	2	0	0
Canker	0	22	0	0
Fresh	0	0	32	0
Greening	0	0	0	22

Furthermore, it must be noted that while the average testing accuracy after background standardization (Table 12) reached 98.45%, the aggregated confusion matrix for the final ensemble model (Table 13) shows an accuracy of 97.96%. This slight variation is attributed to the majority voting mechanism of the ensemble, which prioritizes consensus across folds, and the mathematical difference between averaging individual fold metrics vs. calculating metrics from the pooled global predictions.

To further evaluate whether the observed performance improvements after background standardization are statistically significant, we conducted paired t-tests on fold-wise cross-validation results (before vs. after preprocessing). The results indicate that the differences in accuracy and other evaluation metrics are not statistically significant at a 95% confidence level ($p > 0.05$). This confirms that while the improvements are consistent, they remain modest in magnitude and should not be interpreted as statistically significant gains in predictive performance. Instead, the primary benefit of background standardization lies in improving explanation reliability rather than achieving substantial accuracy improvements.

To further assess the statistical reliability of the proposed model, we performed one-sample t-tests based on the 5-fold cross-validation results. Specifically, we tested whether the obtained performance metrics significantly exceeded a strong baseline threshold of 0.8, which is commonly considered a high-performance benchmark for classification tasks. As shown in Table 14, all evaluation metrics, including accuracy, precision, recall, and F1-score, are significantly higher than this threshold ($p < 0.001$). This indicates that the proposed DeepOrangeNet consistently achieves strong and statistically reliable performance across all folds.

3.5.3 LIME-Based Explainability Analysis after Background Standardization

To complement the quantitative evaluation, we analyze model behavior using LIME explanations after background standardization. After training the proposed DeepOrangeNet model on the pre-processed dataset, we saved the best-performing model and used LIME on the testing set to generate explanations. The aim is to analyze whether background standardization will result in a more meaningful focus on the most important regions.

Table 14: Statistical significance tests for model performance.

Test Type	Comparison	*p*-Value	Significance
One-sample *t*-test	Accuracy > 0.8	<0.001	✓
One-sample *t*-test	Precision > 0.8	<0.001	✓
One-sample *t*-test	Recall > 0.8	<0.001	✓
One-sample *t*-test	F1-Score > 0.8	<0.001	✓

After background standardization (Fig. 6), LIME focuses on disease-relevant regions rather than background artifacts. However, as quantified in Table 15, LIME still does not consistently capture fine-grained pathological features. Most importantly, as background standardization progresses, the emphasized areas tend to overlap with meaningful biological disease characteristics, rather than fruit general body parts. In other words, with canker, LIME tends to focus more on localized regions containing disease "lesions" rather than uniform regions of peel background. In black spot, LIME explanations are centered around dark and depressed regions, and with greening, emphasis is shifted to irregularly colored regions rather than the background.

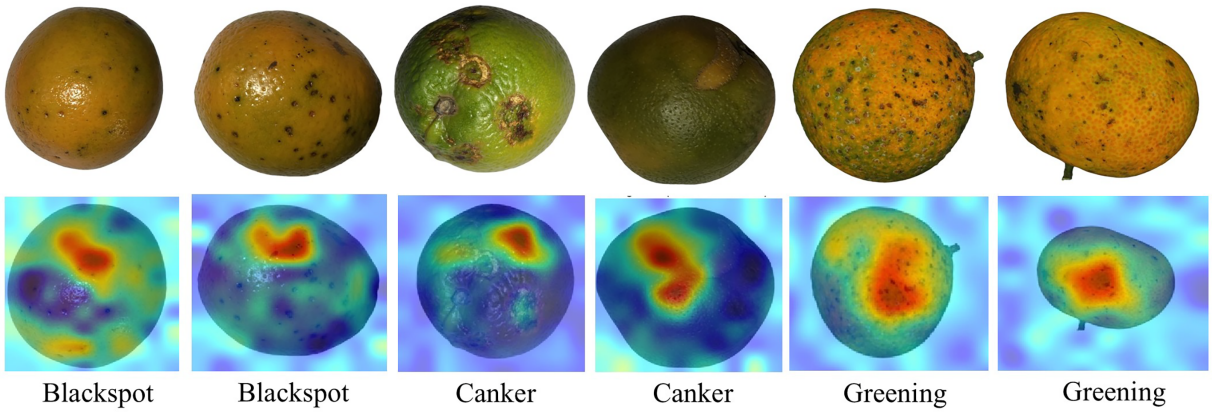

Figure 6: LIME explanation relevance after background standardization. The heatmaps demonstrate a consistent shift in model focus toward the pathological features of the orange fruit (Blackspot, Canker, and Greening) rather than the background. Columns 1–2: Blackspot; Columns 3–4: Canker; Columns 5–6: Greening.

Table 15: Quantitative analysis of LIME explanation relevance before and after background standardization for each disease class. Here, # denotes the number of samples (images) and % denotes the percentage of the total class size. "Irrelevant Focus" specifically refers to instances where LIME identified background regions as important instead of targeting the orange regions themselves.

Class Name	Total Samples Checked	# Irrelevant Focus (Before)	# Irrelevant Focus (After)	% Irrelevant (Before)	% Irrelevant (After)	% Improvement
Canker	200	169	15	84.5%	7.5%	77.0%
Blackspot	200	151	18	75.5%	9.0%	66.5%
Greening	200	53	13	26.5%	6.5%	20.0%
Total/Average	600	373	46	62.2%	7.7%	54.5%

Although LIME shifts focus toward more meaningful regions after preprocessing, it does not consistently highlight the most critical disease-specific features, such as distinct lesion patterns or subtle discolorations. This limitation arises because LIME operates on local perturbations and superpixel approximations, which may not align precisely with disease-specific pathological characteristics such as lesion boundaries, texture variations, or fine color transitions. As a result, while LIME improves interpretability at a coarse level, its explanations should be interpreted with caution in expert-level agricultural diagnosis, where precise localization of disease symptoms is critical. This implies that although background standardization improves explainability and model performance, more explainability technique improvement would be necessary to guarantee that models concentrate on the most pertinent characteristics for categorization.

To quantitatively verify the improvement in the relevance of LIME explanation, we visually inspected 600 LIME explanation maps across three classes of diseases, Canker, Blackspot, and Greening, both before and after standardizing the background. The images used in this phase are unseen holdout test set samples. Table 15 summarizes the findings, with a significant reduction in background-centered explanations after standardization. The proportions of the irrelevance focus maps averaged 62.2% and reduced to 7.7%, achieving a 54.5% improvement. It is of prime interest that this improvement is consistent across all classes, with the main improvement in the Canker class being the improvement 77%. The comparatively better LIME performance for the Greening class, both before and after standardization, is a result of the consistent black background in the original images and the uniform white backgrounds after processing, which together minimized background distraction compared to the more variable backgrounds (of original dataset images) in the other classes. These result provides quantitative support for improved explanation reliability, which is particularly important for real-world agricultural diagnostics where visual interpretability is critical.

Importantly, the magnitude of improvement in explainability relevance (54.5% reduction in background-focused LIME maps) substantially outweighs the numerical gains in classification accuracy, emphasizing that the primary benefit of background standardization lies in explanation fidelity rather than raw performance improvement. These results confirm that background standardization is the primary factor driving improvements in explanation reliability, while the role of the architecture is mainly to provide a consistent predictive framework.

These findings also clarify that the primary source of improvement observed in this study is not only because of the architectural modifications in DeepOrangeNet but is also largely driven by the background standardization process. While the proposed architecture provides a stable and effective classification framework, the substantial reduction in irrelevant LIME focus demonstrates that preprocessing plays a dominant role in improving explanation quality. In contrast, the observed gains in classification accuracy remain relatively small, further supporting that the key contribution lies in enhancing interpretability rather than significantly altering predictive performance.

Notably, the improvement in explanation reliability is significantly larger than the numerical gains in classification performance. This indicates that the primary benefit of background standardization lies in reducing background bias rather than improving raw predictive accuracy.

These findings also highlight a broader limitation: augmentation-based dataset expansion, especially when derived from a limited number of original samples, may not fully capture real-world variability and can lead to an overestimation of model robustness.

3.5.4 Disease-Correlated Feature Extraction Behavior of DeepOrangeNet

Although we do not introduce a new handcrafted feature descriptor in our proposed DeepOrangeNet approach, its feature extraction strategy is intended to promote disease-relevant representation learning

through multi-scale grouped convolutions, channel shuffling, hybrid activation functions, and a self-attention layer. The usage of large convolutional filters, such as 11 × 11 and 5 × 5, can potentially capture coarse-level structural patterns, including overall lesion distribution, as well as color irregularities. Whereas small filters, such as 3 × 3 and 1 × 1, are used to extract fine-grained local cues, including spot boundaries, texture discontinuities, and localized discoloration. Grouped convolutions and channel shuffling further encourage feature diversity by preventing channel-wise redundancy and forcing cross-group feature interaction.

This can be observed clearly through the analysis of class-wise correlation between the features learned by the network and specific disease classes via LIME-based explainability analysis. As observed visually in Fig. 6 and quantitatively in Table 13, the learned feature representations consistently activate disease-specific regions rather than background areas. For citrus canker, highlighted regions predominantly correspond to localized lesion clusters; for black spot disease, emphasis shifts toward dark and depressed regions; and for greening disease, attention concentrates on irregular color patterns and mottled surface textures. This consistent class-dependent activation behavior indicates that DeepOrangeNet learns discriminative feature representations aligned with visually meaningful disease characteristics, rather than relying on global color or background cues.

Furthermore, the confusion matrix analysis verifies the disease-correlated feature extraction behavior of DeepOrangeNet. The minimal confusion between classes, particularly the complete separation of fresh samples from diseased categories, indicates that the learned feature representations capture disease-specific pathological characteristics rather than relying on superficial cues. The limited confusion observed between the blackspot and canker can be because of the partial visual similarity in localized lesion regions. However, the low misclassification count confirms effective discriminative learning. This quantitative evidence complements the LIME-based qualitative analysis and demonstrates that our model extracts meaningful and class-relevant representations.

These observations demonstrate that our architecture learns to identify disease-specific features on its own. It distinguishes between categories by recognizing unique spatial and textural patterns. This allows the model to accurately differentiate between the various types of infection.

3.6 Comparison of the DeepOrangeNet's Classification Performance with State-of-the-Art Deep Learning Models

This experiment compares the proposed DeepOrangeNet architecture with five widely used DL models, namely GoogleNet [28], MobileNetV2 [29], ResNet18 [30], SqueezeNet [31], and ShuffleNet [32]. These models were selected to provide a representative comparison across different architectural design philosophies, including deep residual networks (ResNet18), inception-based architectures (GoogleNet), and lightweight models optimized for efficiency (MobileNetV2, ShuffleNet, and SqueezeNet). This selection enables a balanced evaluation of the proposed method against both high-capacity and computationally efficient models. We trained and evaluated all models under the same experimental conditions using the augmented and background-standardized orange disease dataset. To enable a clearer class-wise evaluation, the aggregated confusion matrices for each model are presented in Tables 16–20. These matrices allow detailed inspection of inter-class confusion beyond overall accuracy values.

Across all architectures, most samples are correctly classified, confirming stable learning behavior. However, minor differences in class-level discrimination can be observed. For GoogleNet (Table 16), two blackspot samples are misclassified as canker, and one canker sample is misclassified as blackspot. This bidirectional confusion suggests overlapping visual characteristics between these two disease categories.

Table 16: Aggregated confusion matrix (Five-Fold cross-validation) for the Final GoogleNet model (Majority voting). Rows represent true classes and columns represent predicted classes.

True Class	Blackspot	Canker	Fresh	Greening
Blackspot	20	2	0	0
Canker	1	21	0	0
Fresh	0	0	32	0
Greening	0	0	0	22

Table 17: Aggregated confusion matrix (Five-Fold cross-validation) for the Final ResNet18 model (Majority voting). Rows represent true classes and columns represent predicted classes.

True Class	Blackspot	Canker	Fresh	Greening
Blackspot	20	2	0	0
Canker	1	21	0	0
Fresh	0	1	31	0
Greening	0	0	0	22

Table 18: Aggregated confusion matrix (Five-Fold cross-validation) for the Final MobileNetV2 model (Majority Voting). Rows represent true classes and columns represent predicted classes.

True Class	Blackspot	Canker	Fresh	Greening
Blackspot	20	2	0	0
Canker	1	22	0	0
Fresh	0	0	32	0
Greening	0	0	0	22

Table 19: Aggregated confusion matrix (Five-Fold cross-validation) for the Final ShuffleNet model (Majority voting). Rows represent true classes, and columns represent predicted classes.

True Class	Blackspot	Canker	Fresh	Greening
Blackspot	20	2	0	0
Canker	1	21	0	0
Fresh	0	0	32	0
Greening	0	0	0	22

Table 20: Aggregated confusion matrix (Five-Fold cross-validation) for the Final SqueezeNet model (Majority Voting). Rows represent true classes, and columns represent predicted classes.

True Class	Blackspot	Canker	Fresh	Greening
Blackspot	19	3	0	0
Canker	0	22	0	0
Fresh	0	0	32	0
Greening	0	0	0	22

ResNet18 (Table 17) exhibits a similar blackspot-canker confusion pattern but additionally misclassifies one fresh sample as canker. This indicates slightly weaker separability between healthy and diseased samples compared to the other architectures.

MobileNetV2 (Table 18) demonstrates behavior comparable to GoogleNet, with minor confusion between blackspot and canker but no misclassification involving the fresh or greening classes.

ShuffleNet (Table 19) exhibits performance characteristics similar to GoogleNet, with limited confusion between blackspot and canker classes while maintaining correct classification for fresh and greening samples. This indicates that ShuffleNet achieves efficient feature representation despite its lightweight design, although minor inter-class confusion persists for visually similar disease categories.

SqueezeNet (Table 20), while highly parameter-efficient, shows slightly increased misclassification in the blackspot class, with three samples incorrectly predicted as canker. Although it maintains perfect classification for fresh and greening categories, the increased confusion in disease classes suggests that extreme model compression may limit fine-grained feature discrimination.

These observations highlight that while lightweight architectures such as ShuffleNet and SqueezeNet provide efficient alternatives, they may struggle with subtle inter-class variations compared to the proposed DeepOrangeNet.

In contrast, the proposed DeepOrangeNet (see Table 13) reduces unnecessary cross-category errors and maintains stronger class-wise consistency. While a small degree of blackspot-canker confusion remains, DeepOrangeNet avoids misclassification of fresh samples into diseased categories and preserves complete separability for the greening class. Although the numerical differences in overall accuracy among the models are marginal, the confusion matrices indicate that DeepOrangeNet achieves more stable class-level discrimination, particularly for visually similar disease patterns.

Although the numerical differences in overall accuracy among the models are marginal, the confusion matrices indicate that DeepOrangeNet achieves more stable class-level discrimination compared to both conventional architectures and recent lightweight models such as ShuffleNet and SqueezeNet, particularly for visually similar disease patterns.

3.7 Comparison with State-of-the-Art Approaches

We compare the proposed framework with several state-of-the-art techniques reported in the literature. However, these comparisons should be interpreted with caution, as prior studies differ in dataset versions, preprocessing strategies, augmentation protocols, and evaluation methodologies. In particular, variations in dataset splits, the presence of duplicate images in earlier dataset versions, and the absence of standardized validation protocols make direct, one-to-one comparison challenging. Therefore, the results presented in Table 21 provide an investigative rather than strictly controlled comparison of performance across studies.

Table 21: DeepOrangeNet performance comparison with state-of-the-art methods.

Work and Year	Method	Total Number of Images	Explainability Method	Validation Accuracy	Testing Accuracy
[6] 2018	K-mean clustering and multi-class SVM	20	No	90.0%	–
[7] 2020	CNN	68	No	93.21%	–
[12] 2024	MobileNetV2 and ML classifiers	1164 (old version of the same dataset)	No	98.22%	–
[13] 2024	Transfer learning of MobileNetV2	1164 (old version of the same dataset)	No	95.0%	–
[33] 2020	KNN, Random Forest and Multiple SVM	277	No	93.0%	–
[34] 2021	K-mean clustering and multi-class SVM	175	No	82.0%	–
[35] 2023	Inception module with EfficientNetV2 for multi-scale feature extraction	2000	No	95.0%	–
[36] 2024	Transfer learning, tuning a pre-trained DenseNet121 model	1090 (same dataset)	No	96.0%	–
[37] 2024	Transfer learning of MobileNetV2	1090 (same dataset)	No	95.0%	–
[38] 2024	Fine-Tuned MobileNetV2	1090 (same dataset)	No	98.66%	–
[39] 2025	Fine-Tuned MobileNetV2 Model	1090 (same dataset)	No	–	97.0%
[Proposed method] 2024	DeepOrangeNet	1090	LIME	99.60%	98.45%

To address recent advancements, we extended our comparison by incorporating recently published methods in 2024 and 2025 that used the same dataset or its earlier version. As seen from Table 21, even recent works, such as fine-tuned MobileNetV2 models [33–39] and transfer learning-based approaches with DenseNet121 [36] achieved satisfactory classification performance but did not have explainability modules and were predominantly based on lightweight or pre-trained architectures. It is worth noting that the works using the older version of the dataset [12,13] reported higher image counts due to the presence of duplicate images, an issue resolved in the latest version, which explains the reduced yet more reliable image count in our study.

Furthermore, most prior works do not incorporate explicit explainability mechanisms, focusing primarily on predictive performance. While this is appropriate for many classification tasks, it limits their applicability in scenarios where interpretability and transparency are critical, such as agricultural decision support systems. Within the constraints of these experimental differences, DeepOrangeNet achieves competitive performance while additionally integrating explainability through LIME and background bias mitigation. This combination of predictive accuracy and interpretability distinguishes the proposed approach from prior methods that focus solely on classification performance. Furthermore, unlike some prior works that do not explicitly verify dataset integrity, our approach ensures strict separation between training and testing sets, enhancing the credibility of the performance comparison. Such an end-to-end architecture makes DeepOrangeNet a more stable and transparent alternative to newer state-of-the-art models. However, a fully controlled comparison would require re-implementation and evaluation of all baseline methods under identical data splits and preprocessing conditions, which is beyond the scope of this study and represents an important direction for future work.

The proposed DeepOrangeNet model achieved the best results because we used kernels of different sizes in the proposed model, such as 11 × 11, 5 × 5, 3 × 3, and 1 × 1 to obtain both high and low-level features for the classification. The kernel of large size, such as 11 × 11, is employed to incorporate the information with large receptive fields (extract high-level features), whereas a small kernel of size 1 × 1 is employed to extract local and fine-grained features (more detailed features).

Furthermore, to improve transparency and support practical deployment assessment, we report the computational characteristics of DeepOrangeNet, including trainable parameters and inference time in Table 22. These metrics provide additional details about the efficiency of the proposed model and facilitate future comparisons as more studies begin to disclose similar implementation details. DeepOrangeNet contains approximately 2.81M parameters, indicating that high classification performance can be achieved without excessive model complexity.

Table 22: Computational complexity and deployment characteristics of the proposed DeepOrangeNet model.

Metric	Value
Trainable Parameters	2,817,444 (~2.81M)
Model Size	11.09 MB
Average Inference Time	0.0395 s/image
Deployment Suitability	Edge/low-resource devices

3.8 Evaluation on an Independent Orange Leaf Disease Dataset

The core aim of this experiment is to verify the effectiveness and generalization ability of the proposed approach. In this experiment, we used a publicly available dataset, “Orange leaf disease dataset” [40]. We

used 4500 images (900 images of each citrus canker disease leaf orange, citrus nutrient deficiency yellow leaf orange, healthy leaf orange, multiple diseases leaf orange, and young healthy leaf orange) for the training and validation (in a five-fold cross-validation setting). The DeepOrangeFNet framework requires 730 min and 43 s to train. But this time depends on the number of folds (k-fold, i.e., repetitions), epochs and iterations. Additionally, it also depends on the computing power utilized. The DeepOrangeNet underwent 560 iterations (in each repetition) during the training stage—28 iterations each epoch, for 20 epochs in one round of five-fold cross-validation. We achieved the average validation accuracy, precision, recall, and F1-score values of 98.04%, 95.32%, 95.6%, and 95.468%. As indicated in the results (Table 23), the DeepOrangeNet model effectively classifies most of the images during training and validation, employing softmax and classification layers.

Table 23: Five-fold cross-validation results of DeepOrangeFNet using images of "Orange leaf disease dataset".

Metric	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
Average	98.04	95.32	95.6	95.468
Median	99.95	99.8	99.8	99.8
Maximum	100.0	100.0	100.0	100.0
Minimum	90.42	77.4	78.8	78.09
Standard deviation	3.81	8.96	8.40	8.68
Variance	14.5	80.29	70.58	75.51

Importantly, these results are obtained on an independent leaf-based dataset, which differs from the primary fruit dataset in terms of imaging conditions, anatomical structure (leaf vs. fruit), and disease manifestation patterns. This provides evidence that DeepOrangeNet can generalize across different data distributions within the citrus domain. However, since both datasets belong to the same crop category, this evaluation reflects intra-domain generalization rather than true cross-domain generalization. Validation across different crop species and imaging domains remains necessary to fully establish broader generalization capability.

4 Discussion

4.1 Key Findings

The study proposed a lightweight DL framework for accurate and interpretable classification of orange fruit diseases. DeepOrangeNet combines disease-oriented feature extraction, attention mechanisms, and hyperparameters for better classification performance and generalization. Compared to traditional machine learning classifiers, DeepOrangeNet achieved better classification performance in all cases, validating its potential as a standalone classifier and feature extractor.

In addition to the predictive capability, the importance of explainability has been stressed in the research. Through the incorporation of LIME, the research showed the tendency of the traditional models to depend on background features. By addressing the problem in the proposed method, the reliability of the predictions has been improved. From the application perspective, the proposed framework has the potential to be used in precision agriculture. For instance, the framework has the potential to be used in the early detection of diseases and the reduction of the usage of pesticides.

4.2 Deployment Feasibility and Edge Environment Considerations

Although the training process of the proposed DeepOrangeNet model consumed approximately 562 min on a CPU-based development framework (Intel i5 processor with 8 GB RAM), however, this configuration represents a development environment with limited resources rather than the intended deployment setup. Moreover, this training process is a one-time effort. Once the model has been trained, inference (i.e., using the model to classify new images) is significantly faster and computationally more efficient. Specifically, the DeepOrangeNet model takes an average inference time of approximately 0.0395 s per image, making it suitable for real-time or near-real-time applications. Furthermore, in terms of computational complexity, the model averaged 2,817,444 trainable parameters and a memory usage of 11.09 MB, further verifying its lightness and deployability in low-resource environments. Although the original dataset is relatively small, the proposed model is designed to be lightweight and parameter-efficient compared to conventional DL architectures. The use of grouped convolutions and channel shuffling reduces redundant parameters while preserving representational capacity. In addition, overfitting is mitigated through data augmentation, dropout regularization, and controlled hyperparameter tuning. The strong performance on validation and testing sets suggests that the model achieves a good balance between complexity and generalization.

For real-world agricultural application scenarios such as disease diagnosis through Unmanned Aerial Vehicles, smart traps, or IoT sensor monitoring systems, the DeepOrangeNet model trained can be exported to light-weight and optimized deployment frameworks such as TensorFlow Lite (LiteRT) and ONNX Runtime. To enable scalable and responsive deployment, these frameworks support a vast range of hardware platforms from mobile, embedded, and IoT devices to cloud services and edge servers. For instance, frameworks such as the NVIDIA Jetson Nano, commonly used in robotics, drones, and smart cameras, or the Raspberry Pi, a low-cost board popular for IoT and smart agriculture applications, can run the optimized model effectively in field settings. To further enhance training and deployment efficiency, additional optimization techniques, such as quantization and pruning, can further reduce computational requirements without significantly affecting performance.

4.3 Limitations

While the proposed approach demonstrates strong performance and statistical reliability, several limitations should be acknowledged.

Evaluation and statistical limitations. Paired statistical comparisons between DeepOrangeNet and competing models were not conducted, which limits the ability to formally assess the statistical significance of observed performance differences. Additionally, the use of non-stratified 5-fold cross-validation may introduce slight variations in class distributions across folds, which can affect the stability and reliability of validation metrics. Although consistent splits were used for fair comparison, future work will incorporate stratified validation and more rigorous statistical testing.

Data and generalization limitations. The model shows some reliance on homogeneous backgrounds, which raises concerns about robustness under real-world variability. Although augmentation increases the training dataset to 1800 images per class, it is still derived from only 991 original samples. As a result, the augmented data does not introduce fundamentally new real-world variability but instead produces transformed versions of existing images. This may make the model appear more robust and increase the risk of overfitting. Furthermore, real-world conditions such as illumination changes, occlusion, and multiple objects per image are not fully represented. The study is also limited to orange fruit datasets, and although evaluation on a leaf dataset is included, both datasets belong to the citrus domain. Therefore, the observed generalization should be interpreted as a within-domain (intra-citrus) generalization and does not

confirm robustness across different crop types or imaging domains. Further validation on multi-crop and multi-source datasets is required to establish true cross-domain generalization.

Model and computational limitations. The use of grid search for hyperparameter optimization introduces significant computational cost, which may limit accessibility for deployment on low-resource systems. Additionally, although six classifiers are used to evaluate the feature extraction capability of DeepOrangeNet, this selection may not fully represent all possible downstream tasks or model architectures. Some classes also exhibit relatively lower precision, recall, and F1-scores, indicating that further optimization and evaluation are needed for consistent performance across all categories.

Explainability limitations. While LIME improves spatial interpretability and reduces background bias, it primarily provides region-level explanations and does not explicitly model fine-grained biological traits such as lesion morphology, depth, or texture. Consequently, the association between highlighted regions and disease-specific pathological characteristics remains indirect and qualitative, which may limit its diagnostic precision for expert-level agricultural decision-making. Furthermore, despite substantial improvement, 7.7% of LIME explanations still exhibited some background focus (Table 15), indicating that background bias is reduced but not completely eliminated. Inconsistencies indicate occasional over-reliance on background features, as shown by decreased validation accuracy when backgrounds are standardized. Furthermore, more advanced class-discriminative attribution methods such as Grad-CAM++ and Score-CAM were not benchmarked in this study. Although these methods can offer improved localization sensitivity compared to standard Grad-CAM, their exclusion limits a systematic comparison of explanation fidelity across different saliency-based approaches. Finally, relying solely on background standardization is insufficient for accurate explainability, indicating the need for complementary methods.

4.4 Future Directions

In the future, we want to make use of evolutionary algorithms to optimize hyperparameters, including but not limited to learning rate, regularization parameters, and batch size, as well as architectural choices such as activation functions and number and size of the layers, along with optimizing the weights of the layers of the model. Also, the real-world deployment of an efficient model after further optimization in the future will validate its practical applicability. Future work will focus on acquiring and testing the model on more diverse real-world images with variable lighting, occlusions, and more than one fruit per frame. This test will be used to estimate the actual generalization capacity of DeepOrangeNet and allow further optimization for field agricultural pathology diagnostics. Additionally, extending the evaluation to larger and more diverse datasets will further validate the generalizability of the proposed framework.

Future work will include systematic benchmarking against advanced explanation methods such as Grad-CAM++ and Score-CAM to evaluate whether they provide more stable and biologically precise attribution under background variability. Future work will also explore symptom-aware and structure-sensitive explainability techniques that explicitly capture disease-specific visual traits, such as raised lesion contours in citrus canker and depressed necrotic regions in black spot disease. Integrating part-based explanations, shape-aware attention mechanisms, or pathology-guided annotations could further strengthen the biological relevance and practical utility of explainable models in real-world agricultural applications.

5 Conclusions

This study addressed the challenge of background bias in explainable fruit disease classification by proposing DeepOrangeNet, a lightweight DL framework that integrates attention-based feature learning with background-aware preprocessing. The primary contribution lies not in introducing new architectural components but in their effective integration to improve both predictive performance and interpretability.

Experimental results demonstrate that data augmentation and careful dataset preparation enhance generalization, while the proposed background-standardization strategy significantly improves the focus and consistency of explainable predictions. In particular, the integration of LIME with background-aware preprocessing provides clear, pixel-level explanations that emphasize diseased regions and suppress irrelevant background artifacts, addressing a key limitation of existing explainable deep learning approaches in agricultural imaging.

Overall, the framework provides a balanced solution that achieves competitive accuracy, efficiency, and improved interpretability compared to existing methods that lack explainability analysis, making it suitable for real-world agricultural applications. While current validation is limited to controlled datasets, the insights gained from this study are broadly applicable to other crops and imaging conditions. We also note that LIME explanations, despite improvement, do not consistently highlight fine-grained disease-specific features, and further work on explanation methods is needed for agricultural diagnostic applications. Future work will focus on extending the approach to more diverse environments and improving explainability techniques for practical deployment in precision agriculture.

Acknowledgement: The authors would like to thank the Future Artificial Intelligence Research (FAIR) project (PE0000013—CUP B53C22003630006), Spoke 3—Resilient AI, within the National Recovery and Resilience Plan (PNRR) of the Italian Ministry of University and Research (MUR).

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Naeem Ullah, Ivanoe De Falco and Giovanna Sannino; methodology, Naeem Ullah; software, Naeem Ullah; data curation, Naeem Ullah; formal analysis, Naeem Ullah, Javed Ali Khan, Michelina Ruocco, Antonio Della Cioppa, Ivanoe De Falco and Giovanna Sannino; investigation, Naeem Ullah; writing—original draft preparation, Naeem Ullah; writing—review and editing, Naeem Ullah, Javed Ali Khan, Michelina Ruocco, Antonio Della Cioppa, Ivanoe De Falco and Giovanna Sannino; supervision, Giovanna Sannino. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The original dataset "Orange Diseases Dataset", used in this study, is freely available at https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset. The complete source code and implementation details for the experiments reported in this study are publicly available at: https://github.com/Engr-Naeem-Ullah/DeepOrangeNet-Background-Bias-XAI to enable independent verification.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Weldergs GW. Orange fruit disease detection and classification using AI-based techniques. In: Proceedings of the 2024 International Conference on Information and Communication Technology for Development for Africa (ICT4DA); 2024 Nov 18–20; Bahir Dar, Ethiopia. p. 108–13.
2. Goyal DRBJ, Bhandari R. Comprehensive study on orange disease detection: a review. Int J Sci Res (IJSR). 2024;13(1):1678–85. doi:10.21275/sr24127202127.
3. Patel H, Prajapati R, Patel M. Detection of quality in orange fruit image using SVM classifier. In: Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI); 2019 Apr 23–25; Tirunelveli, India. p. 74–8.
4. Kamelia L, Abdul Rahman TK, Nurmalasari RR, Hamdani KK. Citrus tree nutrient deficiency classification: a comparative study of ANN and SVM using colour-texture features in leaf images. Int J Comput Digit Syst. 2024;15(1):153–65.

5. Thilagavathi K, Sharafath MM, Abimanyu S, Naveen K. Disease detection in orange fruit using machine learning techniques. In: Proceedings of the 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA); 2023 Jun 16–17; Coimbatore, India. p. 1–6.
6. Behera SK, Jena L, Rath AK, Sethy PK. Disease classification and grading of orange using machine learning and fuzzy logic. In: Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP); 2018 Apr 3–5; Chennai, India. p. 0678–82.
7. Saha R, Neware S. Orange fruit disease classification using deep learning approach. Int J. 2020;9(2):2297–301. doi:10.30534/ijatcse/2020/211922020.
8. Dhiman P, Kaur A, Hamid Y, Alabdulkreem E, Elmannai H, Ababneh N. Smart disease detection system for citrus fruits using deep learning with edge computing. Sustainability. 2023;15(5):4576. doi:10.3390/su15054576.
9. Momeny M, Jahanbakhshi A, Neshat AA, Hadipour-Rokni R, Zhang YD, Ampatzidis Y. Detection of citrus black spot disease and ripeness level in orange fruit using learning-to-augment incorporated deep networks. Ecol Inform. 2022;71:101829. doi:10.1016/j.ecoinf.2022.101829.
10. Yang G, Xu N, Hong Z. Identification of navel orange lesions by nonlinear deep learning algorithm. Eng Agric. 2018;38(5):783–96. doi:10.1590/1809-4430-eng.agric.v38n5p783-796/2018.
11. Dhiman P, Kukreja V, Manoharan P, Kaur A, Kamruzzaman M, Dhaou IB, et al. A novel deep learning model for detection of severity level of the disease in citrus fruits. Electronics. 2022;11(3):495. doi:10.3390/electronics11030495.
12. Kumar S, Pandey AK, Raghav D, Gupta G, Srivastava V. A deep learning approach for multiclass orange disease classification. In: Proceedings of the 2024 2nd International Conference on Disruptive Technologies (ICDT); 2024 Mar 15–16; Greater Noida, India. p. 184–9.
13. Chauhan S. Leveraging MobileNetV2 for accurate detection and classification of orange diseases: a study on blackspot, canker, and greening citrus. In: Proceedings of the 2024 Global Conference on Communications and Information Technologies (GCCIT); 2024 Sep 27–28; Cairo, Egypt. p. 1–6.
14. Pai DG, Balachandra M, Kamath R. Explainable AI in agriculture: review of applications, methodologies, and future directions. Eng Res Express. 2025;7(3):032202.
15. Wang B, Pei W, Xue B, Zhang M. Explaining deep convolutional neural networks for image classification by evolving local interpretable model-agnostic explanations. IEEE Trans Emerg Top Comput Intell. 2025;9(6):3806–20. doi:10.1145/3449726.3459452.
16. Rother C, Kolmogorov V, Blake A. GrabCut" interactive foreground extraction using iterated graph cuts. ACM Trans Graph. 2004;23(3):309–14.
17. Cristovão J. Orange diseases dataset; 2024. Kaggle. [cited 2026 Jan 1]. Available from: https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset.
18. Buslaev A, Iglovikov VI, Khvedchenya E, Parinov A, Druzhinin M, Kalinin AA. Albumentations: fast and flexible image augmentations. Information. 2020;11(2):125. doi:10.3390/info11020125.
19. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision; 2017 Oct 22–29; Venice, Italy. p. 618–26.
20. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv:1312.6034. 2013.
21. Meyes R, Lu M, de Puiseau CW, Meisen T. Ablation studies in artificial neural networks. arXiv:1901.08644. 2019.
22. Jahromi AH, Taheri M. A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features. In: Proceedings of the 2017 Artificial Intelligence and Signal Processing Conference (AISP); 2017 Oct 25–27; Shiraz, Iran. p. 209–12.
23. Xu Y, Zhu Q, Fan Z, Qiu M, Chen Y, Liu H. Coarse to fine K nearest neighbor classifier. Pattern Recognit Lett. 2013;34(9):980–6. doi:10.1016/j.patrec.2013.01.028.
24. Wu J, Yang H. Linear regression-based efficient SVM learning for large-scale classification. IEEE Trans Neural Netw Learn Syst. 2015;26(10):2357–69. doi:10.1109/tnnls.2014.2382123.

25. Zhao W, Chellappa R, Nandhakumar N. Empirical performance analysis of linear discriminant classifiers. In: Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231); 1998 Jun 23–25; Santa Barbara, CA, USA. p. 164–9.
26. Shevade SK, Keerthi SS. A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics. 2003;19(17):2246–53. doi:10.1093/bioinformatics/btg308.
27. Zelinsky A. Learning OpenCV—computer vision with the OpenCV library (Bradski, G.R. et al.; 2008) [On the Shelf]. IEEE Robot Autom Mag. 2009;16(3):100. doi:10.1109/MRA.2009.933612.
28. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015 Jun 7–12; Boston, MA, USA. p. 1–9.
29. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–22; Salt Lake City, UT, USA. p. 4510–20.
30. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA. p. 770–8.
31. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: alexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. arXiv:1602.07360. 2016.
32. Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–22; Salt Lake City, UT, USA. p. 6848–56.
33. Peter V, Khan MA, Luo H. Automatic orange fruit disease identification using visible range images. In: Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019; 2019 Nov 16–17; Guangzhou, China. Revised Selected Papers 11. Vol. 11. Berlin/Heidelberg, Germany: Springer; 2020. p. 341–59.
34. Mojumdar MU, Chakraborty NR. Orange & orange leaves diseases detection using computerized techniques. In: Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT); 2021 Jul 6–8; Kharagpur, India. p. 1–4.
35. Huang Z, Jiang X, Huang S, Qin S, Yang S. An efficient convolutional neural network-based diagnosis system for citrus fruit diseases. Front Genet. 2023;14:1253934. doi:10.3389/fgene.2023.1253934.
36. Kaushik P, Sharma P. Automated detection and classification of orange diseases using DenseNet121: a deep learning approach. In: Proceedings of the 2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG); 2024 Feb 16–17; Indore, India. p. 1–6.
37. Kaushik P, Sharma P. Deep learning-based detection of orange diseases using MobileNetV2 for enhanced agricultural diagnostics. In: Proceedings of the 2024 13th International Conference on System Modeling & Advancement in Research Trends (SMART); 2024 Dec 13–14; Moradabad, India. p. 28–33.
38. Singh G, Guleria K, Sharma S. A fine-tuned MobileNetV2 deep learning model for citrus fruit disease classification. In: Proceedings of the 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS); 2024 Oct 9–11; Tashkent, Uzbekistan. p. 418–23.
39. Rani R. Efficient citrus disease classification using a fine-tuned MobileNetV2 model for sustainable agriculture. In: Proceedings of the 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI); 2025 Jan 14–15; Greater Noida, India. p. 1375–9.
40. Basak SK. Orange leaf disease dataset. Kaggle; 2025 [cited 2026 Jan 1]. Available from: https://www.kaggle.com/dsv/13837836.