



Article

Lightweight CNN-Based Computer Vision for Early Detection of *Monilinia* spp. and *Taphrina deformans* in Peach Crops Under Real Field Conditions

Paola Andrea Mateus Abaunza ^{1,†} , Sandra Milena García Ávila ^{1,*,†} , Luisa Paola Zúñiga Castro ^{1,†} ,
Leyiber Orlando Villa Pinto ^{1,†} , Daniel Felipe Silva Gaona ^{1,†} and Ricardo Alirio González Bustamante ^{2,†}

¹ Escuela de Ciencias Básicas, Tecnología e Ingeniería (ECBTI), Universidad Nacional Abierta y a Distancia (UNAD), Calle 14 Sur #14-23, Bogotá 110231, Colombia; paola.mateus@unad.edu.co (P.A.M.A.); lpzunigac@unadvirtual.edu.co (L.P.Z.C.); lovillap@unadvirtual.edu.co (L.O.V.P.); dfsilvag@unadvirtual.edu.co (D.F.S.G.)

² Universidad ECCI, Cra 19 #49-20, Bogotá 111321, Colombia; rgonzalezb@ecc.edu.co

* Correspondence: sandra.garcia@unad.edu.co

† These authors contributed equally to this work.

Abstract

Brown Rot (*Monilinia* spp.) and Leaf Curl (*Taphrina deformans*) are principal fungal diseases affecting peach (*Prunus persica* L. Batsch) production, lacking validated AI-based diagnostic tools in tropical highland orchards. This study presents a compact Convolutional Neural Network (3658 trainable parameters), applied identically to fruit and leaf classification, integrated with a background-removal preprocessing pipeline and evaluated through stratified 5-fold cross-validation on 800 in-situ images from four orchards in Cóbbita and Choachí, Colombia. The proposed architecture achieved mean accuracies of 82.8% (fruit) and 95.3% (leaf), with AUC values of 0.87 and 0.98, and a trained model footprint of approximately 100 KB, supporting storage- and bandwidth-efficient deployment. Benchmarked against ImageNet-pretrained MobileNetV3-Small and MobileNetV2 under an identical protocol, the proposed architecture matched or exceeded MobileNetV3-Small on leaf classification despite a 257-fold smaller parameter count, and achieved comparable or lower inference latency than both larger backbones. To our knowledge, this is the first validated system for simultaneous detection of both pathogens in *Prunus persica* under real field conditions, combining a compact, deployment-ready architecture with an ablation-verified preprocessing pipeline. The proposed model was deployed in the DurAPP web platform, giving peach growers in tropical highland regions a practical, low-footprint diagnostic tool suited to smallholder farming conditions.

Keywords: disease detection; computer vision; convolutional neural network; precision agriculture; *Prunus persica*; *Monilinia* spp.; *Taphrina deformans*; deep learning; peach disease; smallholder farming



Academic Editors: Saeid Jafari,
Kuan-Chen Cheng and Yu Wen Ting

Received: 22 July 2026

Revised: 25 August 2026

Accepted: 27 August 2026

Published: 2 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Plant diseases represent one of the principal threats to global food security, accounting for yield losses between 20% and 40% of total agricultural production worldwide [1,2]. Peach (*Prunus persica* L. Batsch), cultivated across temperate, subtropical, and tropical highland regions, is particularly vulnerable to two economically significant fungal diseases: Leaf Curl (*Taphrina deformans* (Berk.) Tul.) and Brown Rot (*Monilinia* spp.) [3,4]. Leaf Curl

distorts emerging leaves, reducing photosynthetic capacity and limiting plant development; in heavily infected orchards it can affect up to 30% of trees [5]. Brown Rot (*Monilinia fruticola*, *M. fructigena*, and *M. laxa*) initially colonizes flowers and subsequently spreads to fruits, causing rapid tissue decomposition and, under conducive environmental conditions, losses exceeding 80% of production [6,7]. Both pathogens are favored by humid conditions during critical phenological stages and are particularly difficult to manage once symptoms become visible, underscoring the importance of early detection [8].

In Colombia, peach production is strategically concentrated in the department of Boyacá, which accounts for 37.79% of national output. In 2022, the region registered 813.69 hectares under cultivation and produced 11,830.05 tons [9]. Despite this economic importance, recurrent outbreaks of Brown Rot and Leaf Curl continue to depress production volumes and fruit quality, with direct consequences for smallholder farmers' income [10]. The agroclimatic characteristics of tropical highland orchards (including fluctuating temperatures, high relative humidity, heterogeneous topography, and cultivar-specific symptom expression) create conditions that favor disease development while simultaneously complicating timely visual diagnosis [11].

Traditional disease management in peach orchards relies on periodic visual inspection by field technicians, a method that is inherently subjective, labor-intensive, and impractical at the scale required for early intervention. Under real field conditions, variability in natural illumination, background clutter, canopy occlusion, and growth stage further reduces the consistency of manual assessments [12,13]. In smallholder farming systems, where access to agronomic advisory services is limited and economic margins are narrow, the absence of reliable early-detection tools translates directly into delayed responses, excessive fungicide applications, and preventable crop losses [14].

Artificial intelligence (AI) and computer vision have demonstrated strong potential for automated phytosanitary diagnosis by extracting discriminative visual features from digital images [15,16]. Convolutional Neural Networks (CNNs) have become the dominant architecture for image-based plant disease classification, with pretrained models such as VGG19, ResNet50, and EfficientNetB7, applied through transfer learning, consistently achieving classification accuracies above 95% on benchmark datasets [17–20]. More recently, capsule networks [21], transformer-based architectures [22], and hybrid attention mechanisms [23] have further advanced diagnostic performance under controlled conditions. Studies across diverse crop systems (including tomato [24–27], potato [28], coffee [29], onion [30], cassava [31], cucumber [32], and multiple fruit species [33–35]) have validated CNN-based systems achieving accuracies between 87% and 99%, establishing deep learning as the standard approach for automated phytosanitary screening [36].

The application of machine learning and computer vision extends across a wide range of crop species and pathological contexts. Systems developed for oil palm [37–40], date palm [8,41,42], and other perennial crops [10,11,43] have demonstrated the generalizability of these approaches beyond annual crops. Image datasets acquired under diverse field and laboratory conditions [42,44], combined with mobile-compatible neural network architectures [40], have further expanded the operational applicability of AI-based diagnostics in resource-constrained agricultural environments [30]. The integration of advanced sensing modalities (including hyperspectral imaging [43]) alongside standard RGB vision approaches has broadened the detection scope, enabling the identification of subtle symptom indicators that may not be apparent in conventional visible-spectrum images.

However, these approaches are frequently constrained by high computational demand, dependence on specialized hardware, and validation restricted to curated datasets under controlled conditions, limiting their applicability for real-time, field-based diagnosis in resource-constrained agricultural settings such as tropical highland orchards. To address these lim-

itations, this study designs a lightweight, field-validated CNN architecture optimized for low-latency inference on standard mobile devices, without reliance on specialized hardware. Table 1 summarizes representative recent studies on CNN-based plant disease detection, situating the present work relative to the crops, methodologies, and field-versus-controlled acquisition conditions reported in the literature.

Table 1. Summary of representative studies on CNN-based plant disease detection. C = controlled conditions; F = real field conditions.

Ref.	Crop	Disease/Task	Methodology	Accuracy	D	Key Contribution/Limitation
[24]	Tomato	Disease recognition	DNN + PLS regression	90.1%	C	DL–statistical hybrid; controlled images only
[29]	Coffee	Biotic stress + severity	ResNet50	95.2%	C	Integrated severity estimation; single species
[30]	Onion	Downy mildew	Deep learning + CAM	74.1–87.2%	F	Interpretable localization; moderate accuracy
[28]	Potato	Disease detection	ERBFNN	98.9%	C	No real-field deployment
[31]	Cassava	Disease detection	Transfer learning + SMOTE	>5% gain	C	Addresses class imbalance; controlled only
[21]	Multiple crops	Disease classif.	CapPlant (Capsule CNN)	93.0%	C	Robust CNN alternative; no field validation
[17]	Olive cultivars	Cultivar ID	VGG19, ResNet50, EfficientNetB7	98.0%	C	Transfer learning benchmark; no disease focus
[33]	Orange, grape, mango, apple	Fruit disease	EfficientNetB3, ResNet50	94.1–99.3%	C	Multi-fruit classification; controlled only
[45]	Multiple crops	Disease classif.	Mob-Res (MobileNetV2 + residual)	97.7–99.5%	C	Lightweight + explainable; PlantVillage only
[46]	Tomato	Disease + pest	Faster R-CNN, YOLO	>90%	F	Real-field review; highlights data scarcity gap
[47]	Grape, apple, carrot	Disease in field	Smart camera + DL	mAP > 80%	F	Real-field hardware; specialized equipment
[48]	Crops and weeds	Segmentation	Deep semantic segm.	—	C	Crop/weed discrimination; no disease focus
[49]	Multiple crops	Pest identification	Deep residual learning	88.2%	C	Pest ID in complex backgrounds; no disease
[50]	Date fruits	Maturity estimation	Deep CNN + computer vision	97.5%	C	Multi-attribute estimation; not pathology-focused
[51]	Apple	Flower detection	Deep CNN	94.7%	F	Real orchard detection; phenology focus
[52]	Canola	Weed detection	Max. likelihood + CNN	91.0%	C	Weed detection; no disease classification
This study	Peach (<i>P. persica</i>)	Brown Rot (<i>Monilinia</i> spp.) + Leaf Curl (<i>T. deformans</i>)	Lightweight custom CNN + preprocessing pipeline	82.8% fruit; 95.3% leaf (5-fold CV)	F	First simultaneous <i>Monilinia</i> + <i>T. deformans</i> detection in <i>P. persica</i> under real field conditions; DurAPP mobile deployment

The main contributions of this study are:

- To our knowledge, the first validated system for the simultaneous detection of *Monilinia* spp. (Brown Rot) and *Taphrina deformans* (Leaf Curl) in *Prunus persica* under real field conditions.
- A reproducible, multi-stage preprocessing pipeline (background removal, RGB-based contour detection, and uniform resizing) tailored to heterogeneous field-acquired images.
- A compact, unified CNN architecture (3658 trainable parameters, identical for both tissue types) evaluated via stratified 5-fold cross-validation ($F1 = 0.83$ fruit; 0.95 leaf), with its accuracy–efficiency trade-off explicitly quantified, rather than assumed, through direct benchmarking against two established ImageNet-pretrained lightweight/mobile baselines (MobileNetV3-Small and MobileNetV2) under an identical protocol.
- Real-world deployment of the validated models in the DurAPP web application, demonstrating end-to-end feasibility from data acquisition to a field-ready diagnostic tool.

2. Materials and Methods

The development of the diagnostic framework for Leaf Curl (*Taphrina deformans*) and Brown Rot (*Monilinia* spp.) followed a systematic methodology comprising data acquisition, multi-stage preprocessing, model development and validation, and field deployment. This structured pipeline was engineered to maintain diagnostic integrity under heterogeneous agricultural conditions by synthesizing computer vision and deep learning protocols. The methodology prioritizes the technical synergy between image processing and classification algorithms, ensuring that the resulting tool addresses the stochastic variability of in-situ environments. The complete procedural workflow is illustrated in Figure 1.

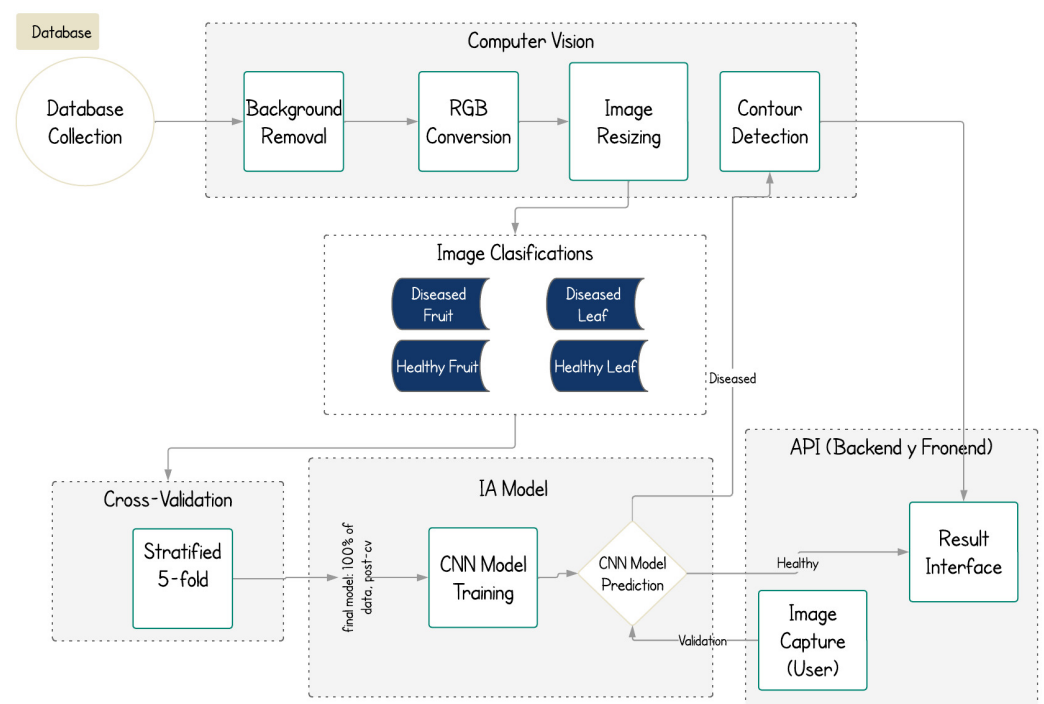


Figure 1. Block diagram of the complete methodology used for the project, from in-situ data acquisition to DurAPP web deployment.

2.1. Study Area and Dataset Acquisition

The dataset was curated from 800 images acquired in-situ across four commercial peach orchards distributed across two geographically distinct regions of Colombia: two orchards located in C6mbita, Boyac6 (approximately 5°42' N, 73°22' W; 2600 m a.s.l.), and two orchards

located in Choachí, Cundinamarca (approximately 4°32' N, 73°55' W; 1800 m a.s.l.), with each orchard covering approximately 1.5 hectares. Acquisition dates were confirmed via EXIF metadata recovered from the original image files and correspond to three production seasons in different years: 6 January 2022 ($n = 412$), 28 April 2024 ($n = 197$), and 31 January 2026 ($n = 191$). Images were captured using smartphone cameras (the modality available to small-holder farmers), ensuring the dataset reflects realistic acquisition conditions and facilitates higher generalization compared to models trained on laboratory-controlled datasets [53]. Images were acquired using two smartphone models operated by the data-collection team: a Xiaomi Redmi Note 9S (Xiaomi Corporation, Beijing, China; f/1.8 aperture, 48 MP sensor; 609 images) and a realme C63 (realme, Shenzhen, China; 191 images). Native capture resolution, recovered from each image's EXIF metadata, varied by device and by transfer method between 800×600 and 4000×3000 pixels for the Xiaomi device and up to 3072×4096 pixels for the realme device; all images, regardless of native resolution, were subsequently downsampled to 150×150 pixels during preprocessing (Section 2.2, Step 3) using BICUBIC resampling (Pillow version 11.2.1).

The repository was structured into two specialized subsets: a fruit-specific group and a leaf-specific group. Each subset comprises 400 images with a balanced 1:1 class ratio (200 healthy and 200 diseased samples; *Monilinia* spp. and *Taphrina deformans* respectively) to mitigate statistical bias during model evaluation. The four diagnostic categories and their image distribution are detailed in Table 2. Representative samples illustrating the phenotypic diversity of healthy and diseased tissues are presented in Figure 2.

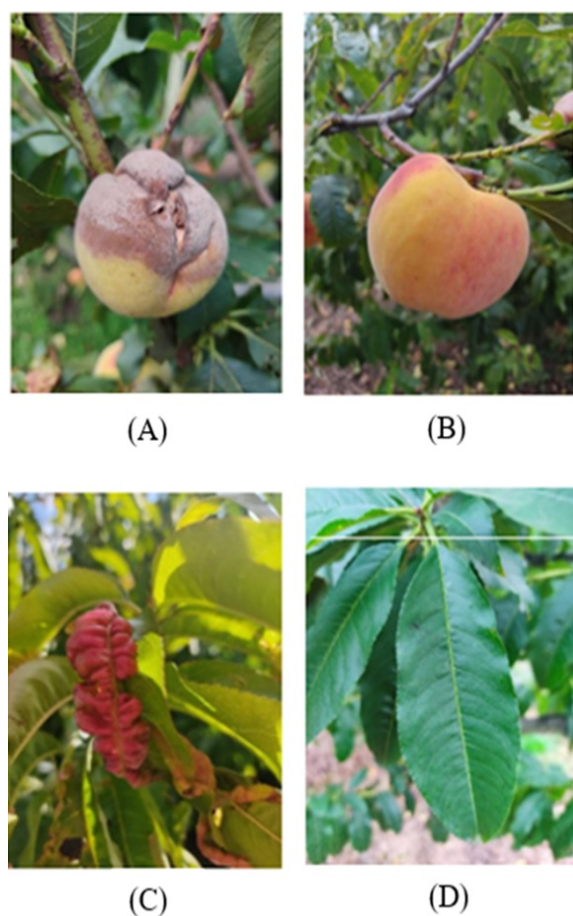


Figure 2. (A) A fruit with Brown Rot (*Monilinia* spp.), showing brown spots and tissue degradation. (B) A healthy fruit for comparison. (C) A leaf with signs of Leaf Curl (*Taphrina deformans*), characterized by deformations and reddish discoloration. (D) A healthy leaf, which serves as a reference for the classification model.

Table 2. All images were acquired in-situ across four commercial peach orchards in C6mbita (Boyac6) and Choach6 (Cundinamarca), Colombia, across three production seasons in different years (6 January 2022; 28 April 2024; 31 January 2026), confirmed via EXIF metadata.

Tissue Type	Health Status	Associated Pathogen	Images (n)
Fruit	Healthy	—	200
Fruit	Diseased	Brown Rot (<i>Monilinia</i> spp.)	200
Leaf	Healthy	—	200
Leaf	Diseased	Leaf Curl (<i>Taphrina deformans</i>)	200
Total			800

2.2. Image Preprocessing Pipeline

Dataset preprocessing constituted a pivotal stage in the computational pipeline, engineered to optimize raw images for feature extraction. This phase systematically mitigated the environmental noise and artifacts inherent to in-situ acquisition while standardizing the input tensors to ensure algorithmic convergence during training. The pipeline comprises four sequential steps implemented in Python 3.10.9.

Step 1: Background Removal. Segmentation algorithms were deployed to isolate fruits and leaves from the dataset, systematically eliminating exogenous background elements such as branches, soil, and overlapping vegetation. Background removal was performed using the rembg library in Python, which applies a pre-trained deep segmentation model to extract the foreground object, ensuring that the feature extraction process focused exclusively on the diagnostic regions of interest and reducing spectral noise from non-target structures [54]. The segmentation results for fruits and leaves are illustrated in Figure 3.

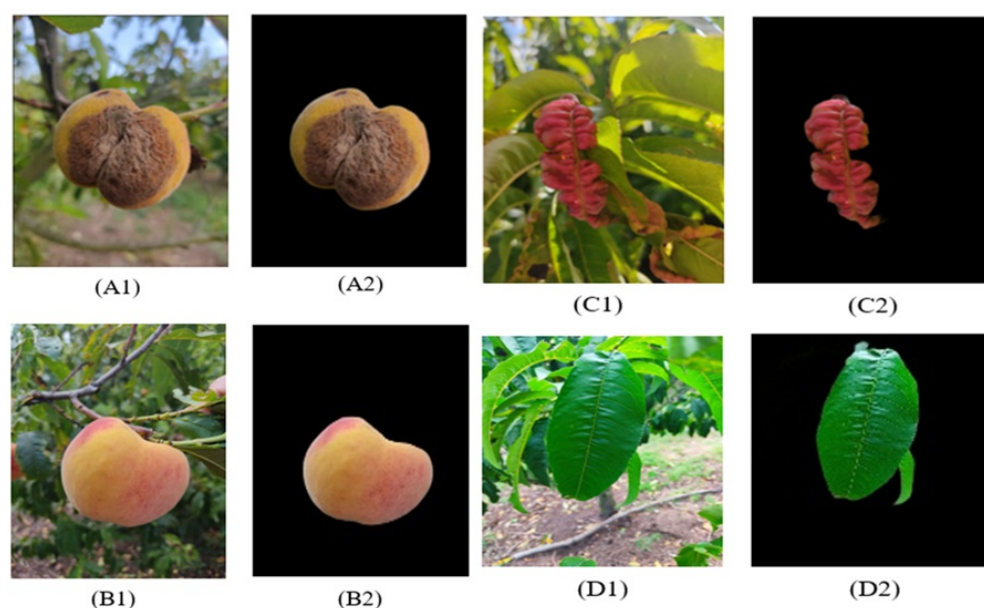


Figure 3. (A1,B1) show diseased and healthy fruits, respectively. (A2,B2) demonstrate background removal on the fruits by segmenting the object of interest. (C1,D1) show a diseased leaf and a healthy leaf, respectively. (C2,D2) demonstrate background removal on the leaves, segmenting the object of interest.

Step 2: Color Space Conversion. Images resulting from the background removal step were saved in PNG format with an alpha (transparency) channel. This RGBA image was converted to the RGB color space using the PIL/Pillow `convert('RGB')` method in Python, which eliminates the alpha channel and ensures chromatic compatibility with TensorFlow/Keras. This conversion is critical for correctly representing the pathological

color indicators: the reddish discoloration of *Taphrina deformans*-infected leaves and the brown necrotic zones of *Monilinia* spp. The converted images were saved in JPEG format. The color space alignment for both fruits and leaves is illustrated in Figure 4.

Step 3: Resizing. Original images, acquired at their native device resolution (varying by device and session; Section 2.1), were resized to a uniform resolution of 150×150 pixels using PIL's `resize()` method with BICUBIC resampling (`PIL.Image.BICUBIC`) to standardize the dataset and optimize computational efficiency. Maintaining consistent input dimensions is imperative for ensuring architectural alignment and stable gradient flow during the learning process, as the CNN requires fixed-size tensors for batch processing [45]. The resulting uniformity in spatial resolution is illustrated in Figure 5.

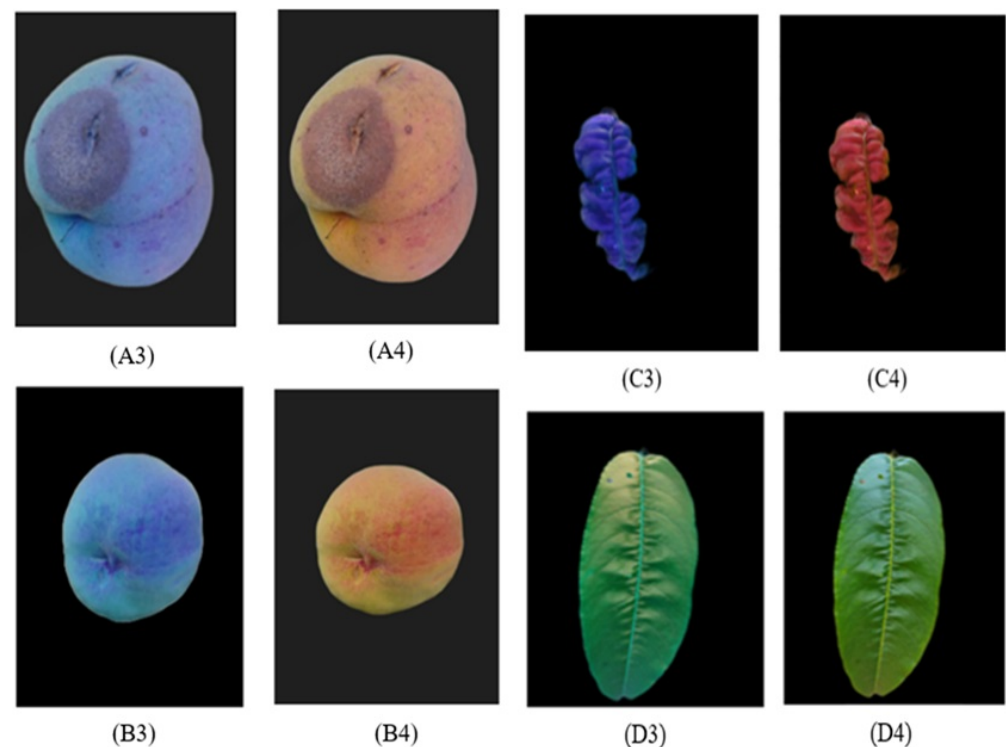


Figure 4. (A3,B3) show a diseased and a healthy fruit in BGR/RGBA format, respectively. (A4,B4) are the versions converted to RGB. (C3,D3) show diseased and healthy leaves in BGR/RGBA format. (C4,D4) correspond to the images converted to RGB.

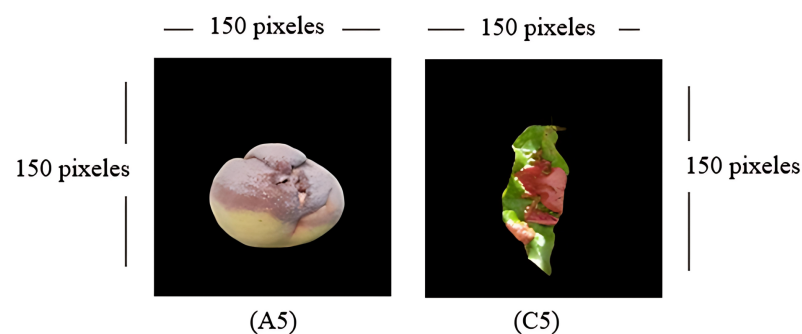


Figure 5. (A5) A fruit with Brown Rot resized to 150×150 pixels. (C5) A leaf with Leaf Curl, resized to the same dimensions.

Step 4: Contour Detection and Image Overlay. A pixel-level color segmentation algorithm was implemented using PIL's `ImageDraw` module to delineate the geometric boundaries and symptomatic perimeters within the dataset. The algorithm iterates over each pixel and evaluates whether its RGB values fall within the color range characteristic

of each pathology. For fruit images, pixels within the brown coloration range of Brown Rot (RGB lower bound: [47, 19, 13]; upper bound: [180, 120, 97]) were highlighted in magenta (255, 0, 255). For leaf images, pixels within the reddish coloration range of Leaf Curl (RGB lower bound: [100, 0, 0]; upper bound: [255, 150, 150]) were highlighted in white (255, 255, 255). These RGB ranges were defined empirically through visual inspection of a subset of symptomatic samples by the authors, rather than through a formal statistical calibration procedure; this manual definition is a design simplification that future work could replace with a data-driven color-range estimation.

It is important to clarify the role of this step in the overall system: the contour-highlighted, color-thresholded image (`superpuesta.jpg` in the deployed codebase) is not an input to the CNN at any stage of training or inference. The classification models were trained and are queried exclusively on the background-removed, resized image produced at the end of Step 3 (`imagen_redimens.jpg`); the contour overlay described in this step is generated separately, after the CNN has already produced its prediction, and is used solely to render the visual explanation shown to the end user in the DurAPP interface (Section 2.9). Consequently, the disease-specific RGB thresholds used to build this overlay do not enter the feature space seen by the classifier and cannot constitute a source of label leakage into model training or into the cross-validation protocol reported in Section 2.6.

The highlighted image was subsequently blended with the original using PIL's `Image.blend()` function ($\alpha = 0.4$ for fruit; $\alpha = 0.6$ for leaf) to produce the superimposed visualization shown to the user. The application of this contouring process on both a peach fruit and a leaf is demonstrated in Figure 6.

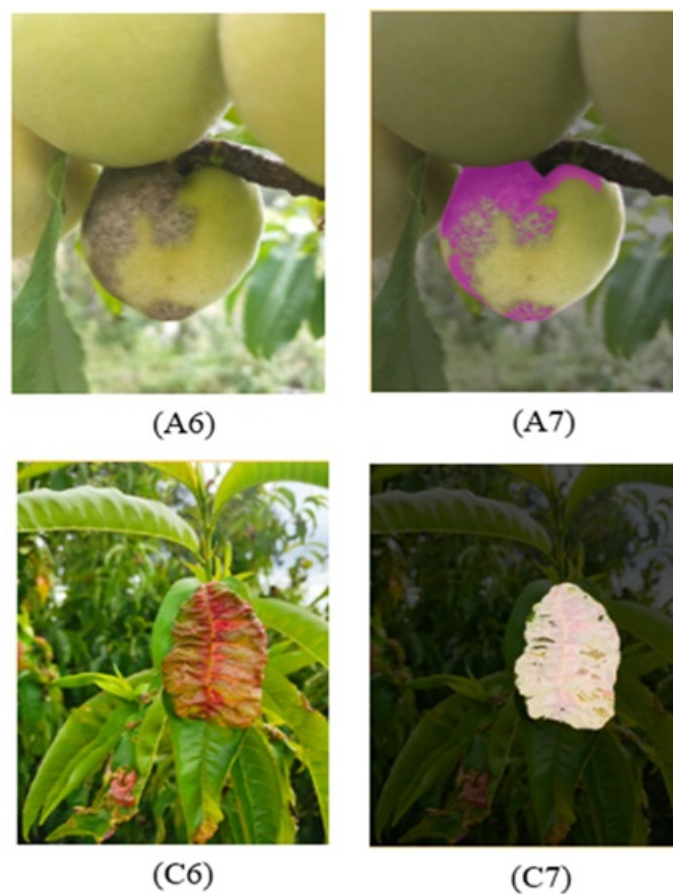


Figure 6. (A6) A fruit with Brown Rot in its original state. (A7) The same fruit with the contour overlaid on the affected area highlighted in magenta. (C6) A leaf with Leaf Curl in its original state. (C7) The same leaf with contour detection identifying the affected area in white.

The preprocessing pipeline also incorporated an EXIF-based orientation correction step. Since images were captured with smartphone cameras that embed orientation metadata, PIL's ExifTags module was used to detect and correct image rotation (90°, 180°, or 270°) prior to any processing step, ensuring consistent spatial orientation across the entire dataset.

The integration of the three preprocessing steps that precede model input (background removal, color space conversion, and resizing) was engineered to amplify feature discrimination while mitigating the stochastic variability inherent in field acquisition; the fourth step, contour detection, is applied downstream of classification and contributes only to the user-facing visual explanation, as clarified above. Regarding data augmentation: standard geometric transformations including rotation, zoom, and horizontal flip were incorporated into the training pipeline (Section 2.5). However, the study prioritized validating diagnostic efficacy using structurally enhanced, in-situ field data rather than relying on synthetic data generation, ensuring that performance metrics reflect genuine field variability [53].

2.3. Dataset Structuring for Model Training

For the model training phase, the dataset was organized into a categorical structure comprising four distinct classes: healthy fruits, healthy leaves, fruits symptomatic of Brown Rot (*Monilinia* spp.), and leaves affected by Leaf Curl (*Taphrina deformans*). The 800 preprocessed images standardized at $150 \times 150 \times 3$ pixels (RGB) with eliminated backgrounds were partitioned into dedicated repositories for leaves and fruits to facilitate specialized feature learning, given that the visual symptom profiles of Brown Rot and Leaf Curl differ substantially in morphology, texture, and color distribution.

2.4. CNN Architecture

A compact Convolutional Neural Network (CNN) was developed for binary classification of healthy versus diseased samples. In contrast to an earlier development iteration that used two separate, non-identical architectures for the fruit and leaf tasks, the architecture reported here is a single design applied identically to both tissue-specific repositories, implemented in TensorFlow/Keras (Python 3.10.9). The design prioritizes two technical dimensions:

- **Spatial Feature Extraction:** The architecture is optimized to capture local spatial patterns and discriminative morphological distortions, such as the distinct lesion margins of Brown Rot and the symptomatic leaf curling induced by *Taphrina deformans*. This localization is critical for achieving high-fidelity classification in plant pathology.
- **Computational Efficiency:** In alignment with the study's focus on field deployability, the design minimizes parameter count through depthwise-separable convolutions and global average pooling in place of a large fully connected layer, the latter being the dominant source of parameters in conventional small CNNs.

The network comprises an input rescaling layer (1/255 normalization) followed by one standard convolutional block and three depthwise-separable convolutional blocks with a 8–16–32–64 filter progression, each followed by ReLU activation; the first three blocks are followed by 2×2 max-pooling. Depthwise-separable convolutions factorize a standard convolution into a per-channel spatial convolution followed by a 1×1 pointwise convolution, substantially reducing parameter count relative to standard convolutions of equivalent filter depth [45]. Feature maps are then reduced via Global Average Pooling rather than flattening into a dense layer, followed by a Dropout layer (*rate* = 0.4) and a final Dense layer with softmax activation producing the binary classification output. The complete layer-by-layer specification, including kernel size, stride, output shape, and parameter count per layer, is reported in Table 3. Trained models were saved in TensorFlow Keras format (.keras).

Table 3. Layer-by-layer specification of the proposed CNN architecture (identical for the fruit and leaf classification tasks). Output shapes omit the batch dimension. Stride refers to the convolution/pooling stride; all convolutions use 3×3 kernels with ‘same’ padding.

Layer	Type	Kernel/Stride	Output Shape	Parameters
Input	–	–	$150 \times 150 \times 3$	0
Rescaling	1/255 normalization	–	$150 \times 150 \times 3$	0
Conv2D	Standard convolution	$3 \times 3/1$	$150 \times 150 \times 8$	224
MaxPooling2D	Max pooling	$2 \times 2/2$	$75 \times 75 \times 8$	0
SeparableConv2D	Depthwise-separable	$3 \times 3/1$	$75 \times 75 \times 16$	216
MaxPooling2D	Max pooling	$2 \times 2/2$	$37 \times 37 \times 16$	0
SeparableConv2D	Depthwise-separable	$3 \times 3/1$	$37 \times 37 \times 32$	688
MaxPooling2D	Max pooling	$2 \times 2/2$	$18 \times 18 \times 32$	0
SeparableConv2D	Depthwise-separable	$3 \times 3/1$	$18 \times 18 \times 64$	2400
GlobalAveragePooling2D	Global average pooling	–	64	0
Dropout	$rate = 0.4$	–	64	0
Dense	Softmax output	–	2	130
Total trainable parameters				3658

This architecture was selected over more complex pretrained models for three specific reasons: (i) the relatively small per-class dataset size ($N = 400$ per model) favors low-complexity architectures less prone to overfitting; (ii) a compact, storage- and bandwidth-efficient model is advantageous when diagnostic tools must be distributed or updated over constrained network connections; and (iii) the binary, task-specific nature of the classification problem does not require the broad feature representations encoded in large ImageNet-pretrained backbones. The quantitative accuracy–efficiency trade-off resulting from this design choice, relative to a pretrained MobileNetV2 backbone, is reported in Section 2.7.

During development, an initial variant of this architecture incorporating Batch Normalization after each convolutional layer was evaluated and discarded: with the batch size used here (16 images) and the limited number of training steps available per fold, the layer’s running statistics failed to stabilize, causing validation accuracy to remain constant at 0.50 despite normally increasing training accuracy. Removing Batch Normalization resolved the issue, and no comparable instability was observed in the architecture reported here.

2.5. Training Configuration and Hyperparameters

Model training was conducted using TensorFlow/Keras. The Adam optimizer was selected for its adaptive learning rate properties, which facilitate stable convergence on small, field-acquired datasets. Categorical cross-entropy was used as the loss function. Training was limited to 10 epochs to prevent overfitting on the relatively small training subsets. The complete set of training hyperparameters is summarized in Table 4.

Table 4. Summary of training hyperparameters and model configuration applied to both fruit and leaf classification models.

Parameter	Value
Optimizer	Adam
Initial Learning Rate	0.001
Loss Function	Categorical Cross-Entropy
Batch Size	32
Training Epochs	10
Input Shape	$150 \times 150 \times 3$ (RGB)
Validation Strategy	Stratified 5-Fold Cross-Validation
Data Augmentation	Rotation, Zoom, Horizontal Flip
Framework	TensorFlow/Keras (Python 3.10.9)
Development Environment	Jupyter Notebook (version 7.4.4)

2.6. Cross-Validation Strategy

Stratified 5-fold cross-validation ($k = 5$) was implemented as the primary evaluation framework to ensure model reliability and minimize overfitting risk. The dataset for each model ($N = 400$) was partitioned into five mutually exclusive subsets, each preserving the 1:1 class ratio through stratified sampling. In each iteration, four folds (320 images) were used for training and one fold (80 images) for independent validation, ensuring every image was used for validation exactly once. For the fruit dataset ($N = 400$), the 5-fold cycle yielded a mean accuracy of 82.8% (95% CI: 81.4–84.1%; $\sigma = 0.94\%$). For the leaf dataset ($N = 400$), a mean accuracy of 95.3% was achieved (95% CI: 92.1–98.4%; $\sigma = 2.29\%$). These metrics, summarized in Table 5, confirm the system's operational viability for real-world phytosanitary diagnosis.

Table 5. Training metrics and stability evaluation via 5-fold cross-validation for both diagnostic models.

Model	Total Images	Diseased (n)	Healthy (n)	Mean Acc. (μ)	Std. Dev. (σ)
Fruit (<i>Monilinia</i> spp.)	400	200	200	82.8%	0.94%
Leaf (<i>T. deformans</i>)	400	200	200	95.3%	2.29%

2.7. Preprocessing Ablation Study

To isolate the contribution of the background-removal step and address the concern that preprocessing choices might drive diagnostic performance independently of disease-specific morphology, the proposed compact architecture (Section 2.4) was retrained and re-evaluated on a second, minimally preprocessed variant of the dataset. In this variant, EXIF orientation correction and RGB color-space conversion were retained, but background removal (Section 2.2, Step 1) was omitted; images were resized directly to 150×150 px without foreground segmentation. Both variants were evaluated under the identical stratified 5-fold cross-validation protocol described in Section 2.6. Note that the contour-detection step (Section 2.2, Step 4) is not part of either variant, as it is not an input to the CNN (see clarification in Section 2.2). Results are summarized in Table 6.

Table 6. Preprocessing ablation: proposed compact architecture with and without the background-removal step, under identical 5-fold cross-validation. Accuracy is reported as mean $\pm$ standard deviation across folds.

Variant	Task	Accuracy	F1-Score	AUC
Without background removal	Fruit	82.8% $\pm$ 3.66%	0.816	0.901
With background removal (full pipeline)	Fruit	82.8% $\pm$ 0.94%	0.830	0.874
Without background removal	Leaf	89.5% $\pm$ 6.00%	0.903	0.971
With background removal (full pipeline)	Leaf	95.3% $\pm$ 2.29%	0.952	0.978

The contribution of background removal differs by tissue type. For fruit classification, mean accuracy is essentially unchanged between the two variants (82.8% in both cases), though background removal reduces the fold-to-fold standard deviation nearly four-fold (0.94% vs. 3.66%) and improves F1-score, suggesting a stabilizing rather than a purely accuracy-driving effect. For leaf classification, background removal provides a clear and consistent benefit, improving mean accuracy by 5.8 percentage points (89.5% to 95.3%) and similarly reducing variance across folds (6.00% to 2.29%). This asymmetry is consistent with the visual characteristics of the two backgrounds encountered in the field: fruit images were typically acquired against comparatively uniform out-of-focus foliage or soil, whereas leaf images were acquired against adjacent leaves and branches of similar color and texture to the leaf of interest, making foreground segmentation more consequential for isolating

the diagnostic region in the leaf task. These results indicate that the reported diagnostic performance is not an artifact of the background-removal step alone, and that its primary contribution is to reduce prediction variance, particularly for the leaf task, rather than to substitute for genuine disease-specific feature learning.

To rigorously and transparently quantify the accuracy–efficiency trade-off of the proposed compact architecture, it was benchmarked against two established lightweight/mobile architectures of graded size, MobileNetV3-Small and MobileNetV2, both with ImageNet-pretrained weights, under conditions matched as closely as possible to the primary experiment. Both backbones (`include_top = False`, ImageNet weights) were kept frozen, and an identical classification head consisting of Global Average Pooling, a Dropout layer (*rate* = 0.3), and a Dense softmax layer was trained on top of each (i.e., standard feature-extraction transfer learning, not fine-tuning of the backbone). All three architectures were evaluated under the identical stratified 5-fold cross-validation protocol, the Adam optimizer, and the identical preprocessing pipeline (background removal, EXIF correction, RGB conversion, and resizing to 150×150 px) applied to the same 800-image dataset. Both pretrained-backbone heads were trained for 10 epochs with a learning rate of 0.0005; the proposed compact architecture was trained for 30 epochs at the same learning rate, reflecting the larger number of steps a substantially smaller model requires to converge with this dataset size. Single-image inference latency was measured for all three final models on a single-vCPU Intel Xeon server instance (2.10 GHz, 4 GB RAM, no GPU, TensorFlow 2.21.0), averaged over 50 repetitions following a 5-iteration warm-up. The benchmarking results are summarized in Table 7.

A fourth candidate, SqueezeNet [55], was also attempted as a comparison point, implemented from its published architecture specification since no ImageNet-pretrained weights for this architecture are distributed with TensorFlow/Keras. Trained from scratch under the same protocol, SqueezeNet failed to converge within a comparable training budget: with the learning rate used for the other models (0.0005), training loss remained flat at $\ln(2)$ (the loss of an uninformative uniform classifier) for the full training run; a substantially lower learning rate (0.0001) allowed loss to decrease but classification accuracy had not moved from chance level after 15 epochs. This is consistent with the general observation that deeper randomly-initialized architectures are difficult to train reliably on datasets of this size without pretrained initialization, as was also observed when attempting to train MobileNetV2 from scratch (Section 4). SqueezeNet is therefore not included in Table 7, since no configuration explored produced a fairly trained model within a comparable budget to the other architectures reported.

Table 7. Proposed compact CNN vs. two ImageNet-pretrained lightweight/mobile baselines (MobileNetV3-Small and MobileNetV2): accuracy, F1-score, AUC, parameter count, and measured single-image inference latency (CPU-only, single vCPU, no GPU) under an identical stratified 5-fold cross-validation protocol on the 800-image dataset. Accuracy is reported as mean $\pm$ 95% confidence interval across the five folds; latency is reported as mean $\pm$ standard deviation over 50 repetitions.

Model	Task	Accuracy (95% CI)	F1-Score	AUC	Parameters	Latency (ms)
Proposed CNN	Fruit	82.8% [81.4–84.1]	0.830	0.874	3658	60.7 $\pm$ 8.3
Proposed CNN	Leaf	95.3% [92.1–98.4]	0.952	0.978	3658	68.1 $\pm$ 25.0
MobileNetV3-Small (TL)	Fruit	91.8% [88.8–94.7]	0.916	0.964	940,274	62.4 $\pm$ 11.4
MobileNetV3-Small (TL)	Leaf	94.5% [92.6–96.4]	0.943	0.991	940,274	68.2 $\pm$ 17.6
MobileNetV2 (TL)	Fruit	94.8% [91.8–97.7]	0.946	0.985	2,260,546	103.7 $\pm$ 5.7
MobileNetV2 (TL)	Leaf	96.8% [94.2–99.4]	0.968	0.996	2,260,546	102.7 $\pm$ 7.3

On leaf classification, the proposed 3658-parameter architecture outperforms MobileNetV3-Small (95.3% vs. 94.5%) despite the latter having 257 times more parameters, and trails MobileNetV2 by only 1.5 percentage points (95.3% vs. 96.8%) despite an

618-fold parameter difference, indicating that additional pretrained capacity does not guarantee proportionally better performance on every task in this domain. Measured inference latency shows a similar pattern of efficiency: the proposed architecture is approximately 40% faster than MobileNetV2 (60.7–68.1 ms vs. 102.7–103.7 ms) and essentially matches MobileNetV3-Small’s latency (62.4–68.2 ms) despite the 257-fold parameter gap, reflecting genuine architectural efficiency rather than a purely parameter-driven effect. On fruit classification, both pretrained backbones reach higher raw accuracy than the proposed architecture (MobileNetV2: 94.8%; MobileNetV3-Small: 91.8%; proposed: 82.8%), consistent with the expected representational advantage of ImageNet pretraining for a task where symptom presentation is more visually variable (Section 4). Taken together, these results show that parameter count alone is an unreliable proxy for either accuracy or latency: the proposed architecture’s efficiency translates into measured gains in footprint and speed, and into competitive or superior accuracy specifically on the leaf task, while the fruit task benefits more from pretrained representations.

The proposed architecture provides substantial, directly measured advantages in leaf-task accuracy, inference latency, and storage footprint, at a fraction of the parameter count of either pretrained backbone (Section 4). On the fruit task, the pretrained low-level filters of MobileNetV3-Small and MobileNetV2, learned from over one million natural images, provide a representational advantage that the proposed architecture, trained from scratch on only 320 images per fold, does not fully match. Both outcomes are reported as measured, without adjustment: together they show that the relevant question for a deployment decision is not which single model is more accurate in isolation, but which combination of accuracy, footprint, and latency best fits the intended constraint, addressed further in Section 4.

2.8. Performance Evaluation Metrics

Model performance was evaluated using a comprehensive set of classification metrics derived from the aggregated confusion matrices across all five cross-validation folds. The following indicators were computed for both models and are reported in Table 8: Accuracy (overall proportion of correct predictions); Precision (proportion of positive predictions that are correct); Sensitivity (Recall) (proportion of actual positive cases correctly identified); Specificity (proportion of actual negative cases correctly identified); F1-Score (harmonic mean of Precision and Recall); and AUC-ROC (area under the receiver operating characteristic curve), computed over the complete 800-image dataset with 80 images per fold for independent testing.

Table 8. Results of the disease detection models: precision, accuracy, sensitivity, and specificity derived from 5-fold cross-validation.

Model	Precision	Accuracy	Sensitivity	Specificity
Brown Rot (<i>Monilinia</i> spp.)	0.817	0.828	0.845	0.810
Leaf Curl (<i>T. deformans</i>)	0.960	0.953	0.945	0.960

2.9. DurAPP Web Application

The validated CNN models were integrated into DurAPP, a web-based application designed for real-time phytosanitary detection under field conditions. The system follows a client–server architecture: the front-end interface was developed using Anvil Works (<https://anvil.works>, accessed on 28 August 2026), a Python-based low-code web development platform, providing cross-platform accessibility from standard mobile devices without specialized hardware. The back-end logic, encapsulating the CNN models (`modelofrutos4.keras` and `modeloTorques.keras`) and the complete preprocessing pipeline, is executed on a dedicated

Python server environment connected to the Anvil front-end via an API uplink key. This client–server decoupling offloads the computational burden from the end-user device to the server, satisfying the low-resource deployment requirements of smallholder farming contexts.

The complete preprocessing sequence (background removal via `rembg`, EXIF orientation correction via `PIL ExifTags`, BGR-to-RGB conversion via `PIL convert('RGB')`, resizing to 150×150 px via `PIL BICUBIC`, and pixel-level contour detection and image blending via `PIL ImageDraw` and `Image.blend()`) is executed server-side prior to CNN inference, making the full pipeline transparent to the end user. The system delivers a dual-modality output: (i) a categorical diagnostic statement identifying the predicted pathology (*Podredumbre Morena* or *Torque*) accompanied by a confidence score expressed as a percentage; and (ii) a visual representation displaying both the original image and a contour-overlaid version that highlights the symptomatic regions used for classification. Given its parameter count (Table 7), the compact architecture is computationally inexpensive to run server-side: measured single-image inference latency was 60.7 ± 8.3 ms (fruit) and 68.1 ± 25.0 ms (leaf) on a single-vCPU Intel Xeon server instance (2.10 GHz, 4 GB RAM, no GPU); the measurement protocol and hardware specification are detailed in Section 2.7. This figure characterizes server-side model inference only and does not include network round-trip time between the user’s device and the DurAPP back-end, which was not measured in this study and depends on the user’s connectivity.

3. Results

The primary objective of this research was to engineer an integrated framework utilizing artificial intelligence and computer vision for the early detection of pathogenic manifestations in peach orchards. To substantiate this objective, a multi-phase developmental and evaluative methodology was executed, focusing on the diagnostic accuracy and operational efficiency of the proposed system. Initially, a high-resolution image repository was curated, comprising both healthy and symptomatic samples of peach fruits and leaves. This dataset specifically targeted the distinct morphological characteristics of Brown Rot (*Monilinia* spp.) and Leaf Curl (*Taphrina deformans*). The resulting categorical database, as illustrated in Figure 7, serves as the foundational corpus for the model, encapsulating a diverse range of healthy and pathological states across both plant tissues.

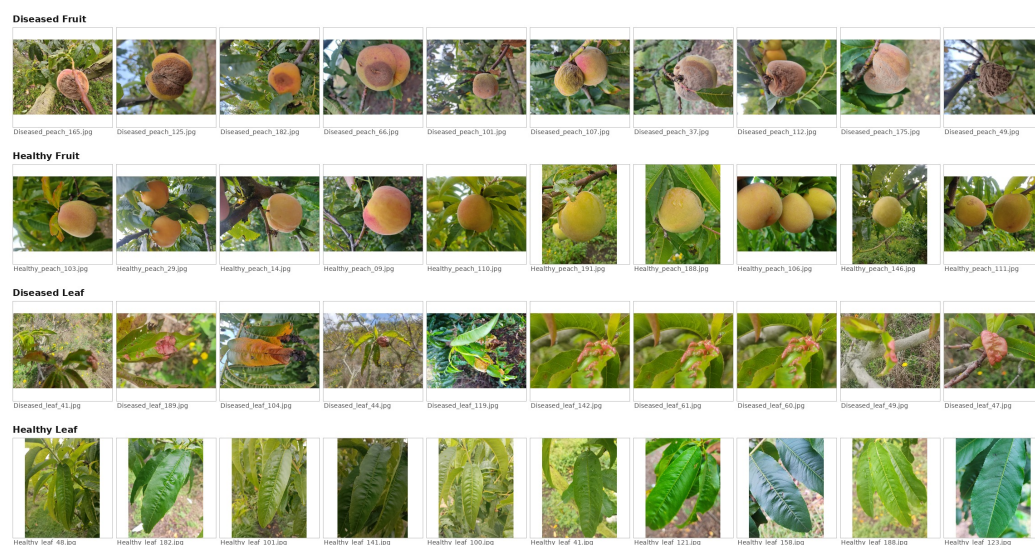


Figure 7. Sample of the curated raw image repository, prior to preprocessing, showing peach fruits and leaves, both healthy and diseased.

Subsequently, computer vision techniques were applied to prepare the images for training the AI model, as seen in Figure 8. Preprocessing techniques, such as background removal, allowed for obtaining the regions of interest in the images.

Subsequently, the convolutional neural network was developed and deployed to analyze the curated image repository and identify pathogenic indicators. During this phase, various analytical techniques were employed to assess the model's diagnostic efficacy. The optimal data partitions for training and testing were established through a 5-fold cross-validation protocol, as illustrated in Figure 9 for both fruit and leaf diagnostic tasks.

The statistical reliability of the models was evaluated through 95% confidence intervals derived from the 5-fold cross-validation protocol. The fruit model demonstrated a mean accuracy of 82.8% (95% CI: 81.4–84.1%), while the leaf model achieved 95.3% (95% CI: 92.1–98.4%). The comparatively low variance observed for the fruit model, and the somewhat wider variance for the leaf model, are discussed in relation to the season-imbalance limitation identified in Section 4.



Figure 8. Same dataset after background removal, isolating the region of interest for fruits and leaves, both healthy and diseased.

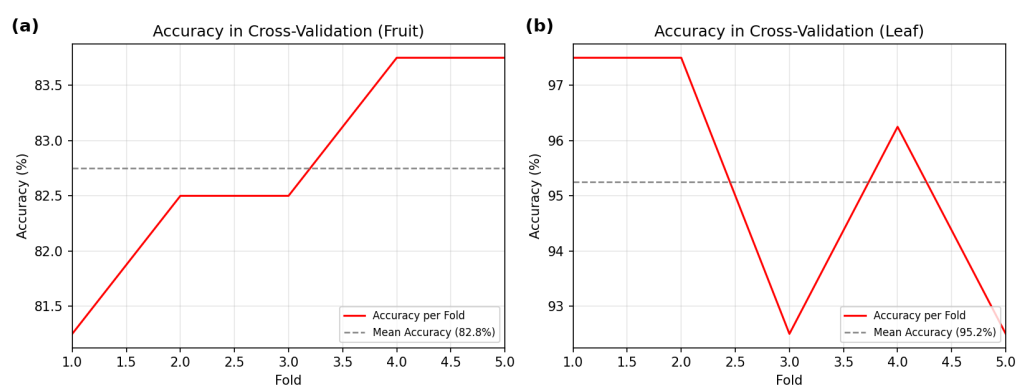


Figure 9. Results of the cross-validation applied to (a) the fruit classification model, achieving a mean accuracy of 82.8%, and (b) the leaf classification model, achieving a mean accuracy of 95.3%.

Regarding the fruit diagnostic task, the cross-validation protocol yielded a mean accuracy of 82.8%, whereas the leaf detection model achieved a mean accuracy of 95.3%. In an earlier development iteration, the final deployed models had been trained on the single fold that yielded the highest validation accuracy for each task; this practice was identified as a source of potential optimistic bias and was discontinued. In the methodology reported

here, the final architectures used for deployment (Section 2.9) are instead trained on the complete dataset (all 400 images per task) only after the full 5-fold cross-validation has been completed and its aggregated metrics reported, so that no test-fold selection influences the final model or the metrics presented in this section. Training dynamics for a representative fold are documented in Figure 10 for both diagnostic tasks.

Finally, the performance of the proposed architectures was evaluated using the aggregated results from the 5-fold cross-validation process. In contrast to a traditional holdout validation, this exhaustive approach ensures that the confusion matrices represent the classification of the entire dataset ($N = 400$ for fruit and $N = 400$ for leaf classification). This methodology provides a statistically robust assessment of the system's diagnostic capabilities by minimizing the impact of data variability.

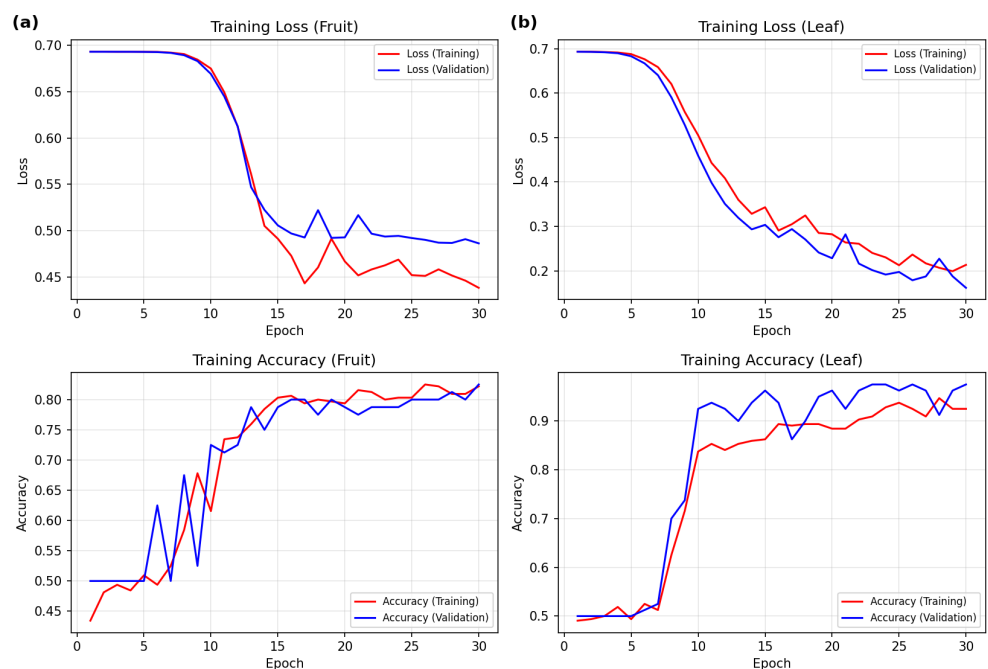


Figure 10. Training accuracy and loss curves for (a) the fruit classification model (*Monilinia* spp.) and (b) the leaf classification model (*Taphrina deformans*).

The evaluation of the fruit classification model yielded an overall accuracy of 82.8% with a balanced performance across both healthy and symptomatic classes. Similarly, the leaf model achieved an accuracy of 95.3%, demonstrating high precision in identifying *Taphrina deformans* manifestations. These aggregated matrices, as presented in Table 9, confirm that the compact CNN architecture maintains high sensitivity and specificity while effectively mitigating the risk of selection bias.

Table 9. Confusion matrices for peach disease diagnostic models ($N = 400$ per task). Results represent the cumulative out-of-fold performance across 5-fold cross-validation iterations. Spec. = Specificity; Sens. = Sensitivity.

Model	Actual Class	Pred. Healthy	Pred. Diseased	Total	Spec.	Sens.
Fruit	Healthy	162	38	200	81.0%	—
	Diseased	31	169	200	—	84.5%
Leaf	Healthy	192	8	200	96.0%	—
	Diseased	11	189	200	—	94.5%

Furthermore, the discriminative capacity of the models was validated through a ROC-AUC analysis utilizing the complete dataset of 800 images. To ensure statistical

reliability, the 5-fold cross-validation was strictly implemented using 80 images per fold for testing. This rigorous validation framework underscores the model's ability to maintain predictive stability across balanced classes within the internal cross-validation setting evaluated here; extending these results to field-based phytosanitary monitoring beyond the orchards and seasons studied will require validation on an independent, external dataset (see Section 4, Limitations).

The discriminative capacity of the proposed models was validated through ROC-AUC analysis utilizing a total dataset of 800 images, balanced equally between healthy and symptomatic classes for both leaf and fruit diagnostic tasks. To ensure statistical reliability, a 5-fold cross-validation protocol was implemented, allocating 80 images per fold for independent testing. As illustrated in Figure 11, the leaf model achieved a mean AUC of 0.98, reflecting high precision in identifying *Taphrina deformans* manifestations. Simultaneously, the fruit model attained a mean AUC of 0.87 for *Monilinia* spp. detection, demonstrating a robust classification threshold despite the significant morphological variability and lighting challenges inherent to fruit samples in field environments.

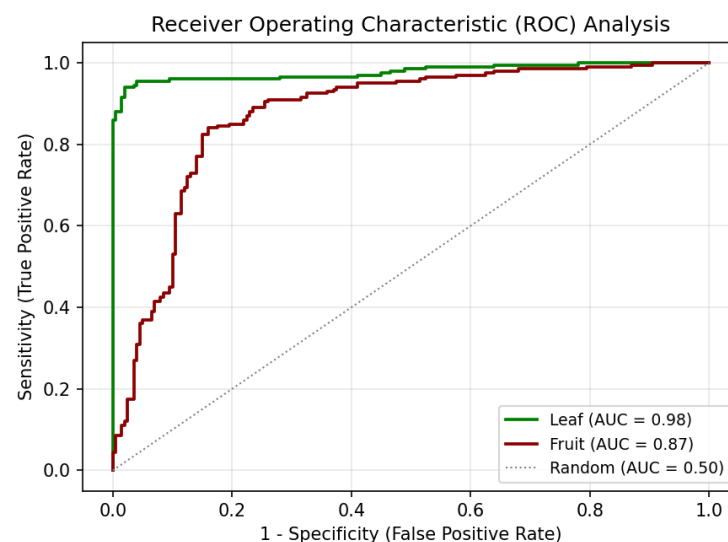


Figure 11. ROC curves and AUC values for the Leaf (*Taphrina deformans*, AUC = 0.98) and Fruit (*Monilinia* spp., AUC = 0.87) models based on 5-fold cross-validation.

To characterize what visual regions drive the models' predictions, and to examine the false positives and false negatives already reported in Table 9 (fruit: 38 false positives, 31 false negatives; leaf: 8 false positives, 11 false negatives), Grad-CAM [56] attention maps were computed for the final layer of each model on a sample of correctly and incorrectly classified test images (Figures 12 and 13). For the leaf model, attention on correctly classified diseased samples concentrates precisely on the curled, discolored lesion area rather than on surrounding healthy leaf tissue, consistent with the model learning disease-specific morphological features rather than incidental background cues. For the fruit model, attention on correctly classified diseased samples is similarly concentrated on the fruit body; however, on the two false-positive healthy samples shown in Figure 12, attention concentrates disproportionately on the outer rim of the segmented fruit silhouette rather than its surface. This pattern is consistent with the fruit model responding, in at least some misclassified cases, to residual artifacts of the background-removal segmentation boundary (Section 2.2, Step 1) rather than to genuine lesion-like surface texture, and is compatible with the preprocessing ablation finding (Section 2.6) that background removal contributed less consistently to fruit classification than to leaf classification. This observation does not overturn the overall diagnostic performance reported above, but it identifies a concrete, visually verifiable candidate explanation for a portion of the fruit model's false positives, and

suggests that refining the background-removal boundary treatment is a specific, actionable target for future improvement of the fruit model.

Regarding the system's integration, adjustments were made to the connection between the back-end and the front-end to achieve optimal image sending and receiving times, adapted to the user's bandwidth. Figure 14 shows the web application interface in operation, illustrating the prediction results for a peach fruit and leaf.

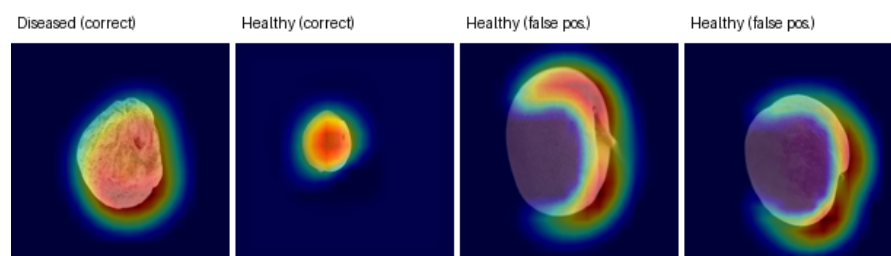


Figure 12. Grad-CAM attention maps for the fruit classification model on four representative test images: a correctly classified diseased sample, a correctly classified healthy sample, and two healthy samples misclassified as diseased (false positives). Warmer colors indicate regions that contributed more strongly to the model's prediction.

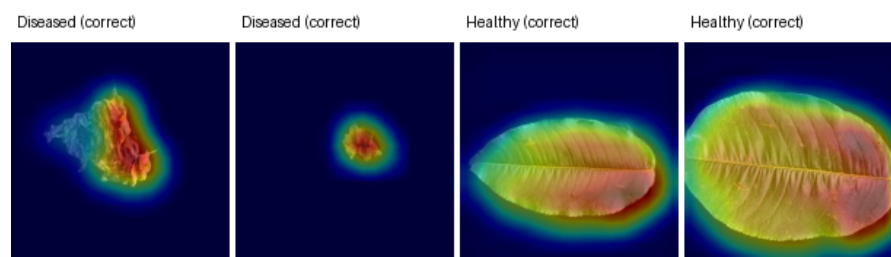


Figure 13. Grad-CAM attention maps for the leaf classification model on four representative, correctly classified test images: two diseased samples and two healthy samples.

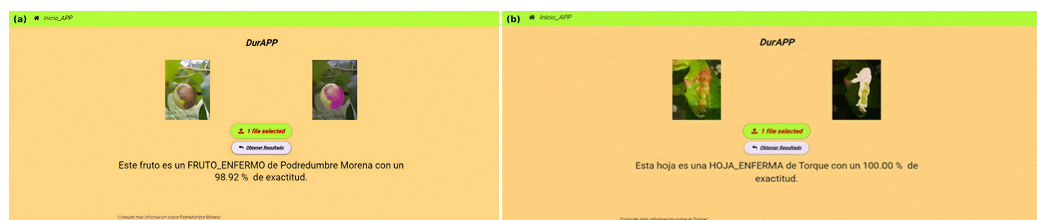


Figure 14. The DurAPP web application interface, showing the diagnostic result for (a) the state of the peach fruit and (b) the state of the peach leaf.

The diagnostic performance of the proposed architectures was evaluated through a multidimensional metric framework to ensure a rigorous assessment of classification reliability. Beyond standard accuracy, the model's efficacy was quantified using Precision, Recall (Sensitivity), Specificity, and the F1-score, all derived from the aggregated true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates. These indicators facilitate an objective analysis of the model's discriminative capacity, particularly regarding its ability to mitigate Type I and Type II errors within a balanced dataset ($N = 800$).

The resulting metrics for both the fruit and leaf classification tasks are synthesized in Table 10. The incorporation of the F1-score, accompanied by its 95% confidence intervals, provides a robust measure of the harmonic balance between precision and sensitivity. This comprehensive analytical approach validates the consistency of the compact CNN architecture, confirming its operational stability across the diverse phenotypic manifestations of *Monilinia* spp. and *Taphrina deformans* in field-acquired data.

Table 10. Performance metrics of the proposed compact CNN models for fruit and leaf disease classification. F1-Score reported as mean with 95% confidence interval across 5-fold cross-validation.

Model Task	Precision	Recall	Specificity	F1-Score (Mean [95% CI])
Fruit Classification	0.817	0.845	0.810	83.0% [81.7–84.4]
Leaf Classification	0.960	0.945	0.960	95.2% [92.0–98.4]

4. Discussion

The results of this study demonstrate the viability of a custom compact CNN architecture for the automated detection of *Monilinia* spp. and *Taphrina deformans* in peach crops under real field conditions. The fruit classification model achieved a mean 5-fold cross-validation accuracy of 82.8% (95% CI: 81.4–84.1%; $\sigma = 0.94\%$), and the leaf model reached 95.3% (95% CI: 92.1–98.4%; $\sigma = 2.29\%$). The higher accuracy of the leaf model is consistent with the symptom profile of Leaf Curl (*Taphrina deformans*), which produces visually distinctive morphological deformations (characterized by pronounced leaf curling, tissue distortion, and reddish-vinous discoloration) that create a highly discriminative visual pattern relative to healthy foliage [3,4]. In contrast, Brown Rot (*Monilinia* spp.) manifests as brown lesions that may overlap visually with other forms of fruit degradation such as bruising, overripening, or abiotic damage under real field acquisition conditions [6,7]. This symptom ambiguity is reflected in the higher false positive and false negative rates of the fruit model (38 and 31, respectively) compared to the leaf model (8 and 11, respectively). The AUC values of 0.98 for the leaf model and 0.87 for the fruit model further validate the discriminative capacity of both architectures, confirming adequate class separation across the complete 800-image dataset under 5-fold cross-validation.

The diagnostic performance achieved in this study must be contextualized within the broader literature on CNN-based plant disease detection. Several published systems report classification accuracies exceeding 95–99%, including CapPlant [21] at 93.0%, ResNet50 applied to coffee leaf disease at 95.2% [29], and fine-tuned CNN-based potato disease detection at 98.9% [28]. However, these results were obtained under controlled laboratory conditions using curated datasets (principally PlantVillage) which lack the environmental variability, background clutter, and heterogeneous lighting inherent to real field acquisition [46,57]. Studies that have evaluated CNN-based systems on real field-acquired datasets consistently report lower accuracy values [45,47,57], confirming that the 82.8% and 95.3% accuracies obtained here represent competitive performance for field-deployed systems.

While the underlying CNN architecture employed here is conventional in its building blocks (standard and depthwise-separable convolutions, max-pooling, global average pooling), and the preprocessing pipeline combines established computer-vision techniques (background segmentation, color-space conversion, geometric augmentation), the three-way benchmarking in Section 2.7 (Table 7) demonstrates several concrete advantages of the proposed compact design. On the leaf classification task, the proposed architecture matched the accuracy of both pretrained baselines and outright exceeded MobileNetV3-Small (95.3% vs. 94.5%) despite using 257 times fewer parameters (3658 vs. 940,274). Measured inference latency was essentially equivalent to MobileNetV3-Small (60.7–68.1 ms vs. 62.4–68.2 ms) and approximately 40% lower than MobileNetV2 (102.7–103.7 ms), showing that the compact architecture is competitive in real-world responsiveness despite its minimal parameter count. The practical storage footprint reflects the same pattern even more starkly: the trained model file occupies approximately 100 KB on disk, versus approximately 4.1 MB for MobileNetV3-Small (42-fold larger) and 9.2 MB for MobileNetV2 (94-fold larger), a difference directly relevant to over-the-air model updates and storage-constrained deployment in the low-bandwidth rural connectivity conditions typical of

the orchards studied here. The architecture is also, unlike the two pretrained baselines evaluated individually per task, a single unified design applied identically to both fruit and leaf classification, simplifying maintenance and future extension of the DurAPP platform.

On the fruit classification task specifically, both pretrained backbones reached higher raw accuracy than the proposed compact architecture (MobileNetV2: 94.8%; MobileNetV3-Small: 91.8%; proposed: 82.8%), a gap consistent with the general expectation that a backbone pretrained on over one million natural images retains a representational advantage over a small architecture trained from scratch on 320 images per fold, particularly for a task such as Brown Rot detection where symptom presentation is visually more variable, as discussed above. This is reported alongside the wins above, rather than in place of them, because the relevant question for a deployment decision is not which single model is accurate in isolation, but which combination of accuracy, footprint, and latency fits a given constraint: for the leaf task, and for any deployment where storage or bandwidth is limited, the results in Table 7 directly favor the proposed compact architecture; for fruit classification where maximal accuracy is the overriding priority and server-side storage is unconstrained, MobileNetV2 remains the stronger choice. We regard the principal contribution of this study, in view of these results, as engineering and applied rather than purely algorithmic: a validated, end-to-end phytosanitary diagnostic system, developed from in-situ field data acquisition across three production seasons (Section 2.1), through a documented and ablated preprocessing pipeline (Section 2.6), to real deployment in the DurAPP platform (Section 2.9), for a disease pair and crop for which, to our knowledge, no validated diagnostic tool previously existed under tropical highland smallholder conditions.

It should also be noted that the validation reported throughout this study is retrospective image classification under cross-validation; no prospective field trial in which growers or extension technicians used DurAPP under real operating conditions to inform actual treatment decisions has yet been conducted. Such a trial would be required to assess practical adoption factors (interface usability, diagnostic turnaround time under field connectivity conditions, and downstream agronomic outcomes) that fall outside the scope of the present classification-focused evaluation, and is identified as a priority direction for future work (Section 5).

The preprocessing pipeline evaluated in Section 2.6 constitutes a documented, quantified contribution of this work rather than an assumed one. Background removal via *rembg* contributed a measurable but tissue-dependent benefit: for fruit classification it left mean accuracy essentially unchanged while reducing fold-to-fold variance nearly four-fold, whereas for leaf classification it improved mean accuracy by 5.8 percentage points and similarly reduced variance, consistent with the greater visual similarity between leaf specimens and their surrounding foliage relative to the typically more distinct fruit-versus-background contrast in the acquired images. The pixel-level contour detection step (Section 2.2, Step 4) is not part of the classifier's input pipeline: it is generated after inference, specifically to render the visual explanation shown to DurAPP end users (Figure 14), and its disease-specific RGB thresholds therefore play no role in the classification metrics reported in this study. The decision to prioritize structurally enhanced, in-situ field data over synthetic augmentation, despite the relatively small dataset size ($N = 400$ per model), ensured that reported performance metrics reflect genuine field variability rather than artificially expanded training distributions [53].

The deployment of DurAPP as a real-time diagnostic tool directly addresses the phytosanitary management needs of peach smallholders in Boyacá and Cundinamarca, where *Taphrina deformans* and *Monilinia* spp. represent the principal limiting factors for production quality and volume [6,9]. The department of Boyacá accounts for 37.79% of national peach production, with 813.69 hectares under cultivation and 11,830.05 tons

produced in 2022 [9]. The dual-modality output, which combines a categorical diagnostic statement with a contour-overlaid visual validation, provides field personnel with both quantitative and visual evidence to support immediate management decisions, minimizing chemical overapplication and contributing to sustainable crop management practices [6,7]. The integration of the validated CNN into the DurAPP web platform via Anvil Works establishes a scalable and maintainable deployment model that can be updated without requiring farmers to install new software.

Several limitations of the present study should be acknowledged. First, although images were acquired across four orchards in two geographically distinct regions, Cóbbita, Boyacá (2600 m a.s.l.) and Choachí, Cundinamarca (1800 m a.s.l.), across three production seasons in different years (Section 2.1), the dataset remains limited to two municipalities within the central Andean region of Colombia. Generalization to other peach-producing zones with markedly different agroclimatic conditions, cultivars, or disease pressure profiles has not yet been validated. Second, class representation is not uniform across the three acquisition seasons reported in Section 2.1. A season-by-season breakdown of the 800-image dataset shows: for leaf images, 60 diseased and 72 healthy samples in the first season, 140 diseased and 0 healthy in the second, and 0 diseased and 128 healthy in the third; for fruit images, 199 diseased and 81 healthy in the first season, 1 diseased and 56 healthy in the second, and 0 diseased and 63 healthy in the third (each season totalling, respectively, 412, 197, and 191 combined fruit and leaf images; see Section 2.1). This pattern reflects the natural condition of the orchards at the time of each field visit rather than a deliberate sampling design: healthy and diseased specimens were photographed opportunistically as encountered, and disease incidence—and therefore the availability of diseased specimens to photograph—varies across visits depending on seasonal disease pressure, rather than being independently controlled per season. As a direct consequence, a season-grouped or orchard-grouped validation scheme (e.g., leave-one-season-out), in which an entire acquisition season is held out for testing, is not statistically meaningful with the current dataset: at least two of the three season-defined subsets contain zero or near-zero examples of one class, so standard classification metrics for that class (recall, precision, F1) would be undefined or computed from a single example in the held-out set. The stratified 5-fold cross-validation reported in Sections 2.6 and 3, which stratifies by class label without controlling for acquisition season, therefore cannot fully rule out that some fraction of the reported diagnostic performance reflects visual characteristics correlated with the acquisition conditions of a given field visit (illumination, camera settings, background vegetation state) in addition to disease-specific morphology. Addressing this will require a future, purpose-designed data collection effort in which healthy and diseased specimens are deliberately sampled in comparable numbers within each season and orchard, enabling a genuine season- or orchard-grouped validation that the present dataset cannot support. Third, the per-class sample size of 200 images, while sufficient to demonstrate diagnostic viability through rigorous 5-fold cross-validation, constrains the statistical power of the evaluation; a larger and more geographically diverse dataset, collected with the class-balancing constraint described above, would yield narrower confidence intervals and more robust performance estimates. Fourth, the current system performs binary classification (healthy versus diseased) within each tissue type and does not distinguish between *Monilinia fructicola*, *M. fructigena*, and *M. laxa* at the species level, which may be relevant for targeted fungicide selection. Fifth, the system currently requires an active server connection for inference; true offline diagnostic capability (desirable for remote orchard locations with limited connectivity) has not yet been implemented. Sixth, “early detection” in this study refers to the identification of visually evident symptomatic stages (before tissue damage compromises fruit or leaf viability) rather than

the detection of subclinical or presymptomatic infection; the current dataset and binary healthy/diseased classification scheme do not include subclinical samples, and extending the system toward presymptomatic diagnosis remains an open challenge, as discussed in Section 5. These limitations do not invalidate the observed trends but suggest that future work should prioritize class-balanced, multi-regional dataset expansion, species-level pathogen discrimination, and edge computing deployment to consolidate the operational applicability of the proposed framework.

5. Conclusions

This study demonstrates the viability of a compact custom CNN framework for the detection of Brown Rot (*Monilinia* spp.) and Leaf Curl (*Taphrina deformans*) in peach crops under real field conditions. The four objectives set out for the work were met: an 800-image dataset was curated in situ across four commercial orchards in C6mbita (Boyac6) and Choach6 (Cundinamarca); a structured preprocessing pipeline was designed and validated; the custom CNN architecture was evaluated through 5-fold cross-validation, reaching mean accuracies of 82.8% for fruit and 95.3% for leaf classification; and the validated models were deployed in the DurAPP web application for field-oriented use. To our knowledge, this is the first validated system for the simultaneous detection of both pathogens in *Prunus persica* under tropical highland conditions, which we regard as the principal scientific contribution of this work.

A second, complementary contribution lies in the quantified efficiency of the proposed architecture: benchmarked against two ImageNet-pretrained lightweight/mobile baselines under an identical protocol, the compact CNN matched or exceeded MobileNetV3-Small on leaf classification (95.3% vs. 94.5%) while using 257 times fewer parameters, and remained within 1.5 percentage points of MobileNetV2 on the same task despite an 618-fold parameter difference; it also achieved lower measured inference latency than both baselines. On fruit classification, both pretrained backbones reached higher raw accuracy, reflecting the representational advantage of ImageNet pretraining for this more visually variable symptom class. Within the conditions evaluated here, this confirms that a task-specific compact CNN is a viable alternative to general-purpose pretrained models where footprint, latency, or over-the-air update costs are limiting factors, while a pretrained backbone remains preferable where maximal accuracy is the overriding priority and server-side compute is unconstrained. Because inference is executed server-side, the practical benefit for end users is that no specialized on-device hardware is required, provided a basic mobile connection to the DurAPP back-end is available. Together with the multi-stage preprocessing pipeline, this supports the broader goal of translating deep learning research into diagnostic tools that smallholder farmers in Boyac6 and Cundinamarca can realistically use in the field.

Future work will follow two complementary lines. The first concerns ecological expansion: collecting data across additional seasons, cultivars, and geographic zones, and evaluating the model under conditions not yet represented in the current dataset, for instance through leave-one-orchard-out validation, to test generalization beyond the orchards studied here. The second concerns technological integration, particularly edge deployment on low-cost hardware (e.g., Jetson Nano units or standard mobile terminals) to enable offline inference, together with species-level pathogen discrimination and the exploration of hyperspectral or multispectral imaging as a path toward detecting subclinical, presymptomatic infection.

Author Contributions: All authors contributed equally to the conceptualization, methodology, validation, investigation, and writing of this manuscript. Software development: L.P.Z.C., L.O.V.P., D.F.S.G., P.A.M.A. and S.M.G.6.; data curation and resources: L.P.Z.C., L.O.V.P., D.F.S.G., P.A.M.A.

and S.M.G.Á.; writing—review and editing: R.A.G.B., P.A.M.A. and S.M.G.Á.; supervision and project administration: P.A.M.A. and S.M.G.Á. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Universidad Nacional Abierta y a Distancia (UNAD) under the Research Call No. 12 (Convocatoria Interna No. 12, 2023), within the project “Identificación de enfermedades en el durazno (*Prunus persica*) mediante el uso de técnicas de visión por computadora e inteligencia artificial”, project code PGI2001ECBTI2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors gratefully acknowledge the Universidad Nacional Abierta y a Distancia (UNAD) for its institutional support throughout the development of this research. The authors also wish to thank the peach growers of Cóbbita (Boyacá) and Choachí (Cundinamarca), for facilitating access to their orchards for in-situ data acquisition.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AUC	Area Under the Curve
BGR	Blue-Green-Red (color space)
CI	Confidence Interval
CNN	Convolutional Neural Network
CV	Cross-Validation
DL	Deep Learning
EXIF	Exchangeable Image File Format
F1	F1-Score (harmonic mean of Precision and Recall)
FN	False Negative
FP	False Positive
IoT	Internet of Things
JPEG	Joint Photographic Experts Group
ML	Machine Learning
PNG	Portable Network Graphics
RGB	Red-Green-Blue (color space)
ROC	Receiver Operating Characteristic
SOTA	State of the Art
TN	True Negative
TP	True Positive
UNAD	Universidad Nacional Abierta y a Distancia

References

1. Savary, S.; Willocquet, L.; Pethybridge, S.J.; Esker, P.; McRoberts, N.; Nelson, A. The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* **2019**, *3*, 430–439. [[CrossRef](#)] [[PubMed](#)]
2. Shoaib, M.; Shah, B.; El-Sappagh, S.; Ali, A.; Ullah, A.; Alenezi, F.; Gechev, T.; Hussain, T.; Ali, F. An advanced deep learning models-based plant disease detection: A review of recent research. *Front. Plant Sci.* **2023**, *14*, 1158933. [[CrossRef](#)] [[PubMed](#)]

3. Rastogi, A.; Arora, R.; Sharma, S. Leaf disease detection and grading using computer vision technology and fuzzy logic. In *Proceedings of the 2nd International Conference on Signal Processing and Integrated Networks (SPIN 2015)*; IEEE: Piscataway, NJ, USA, 2015; pp. 500–505. [\[CrossRef\]](#)
4. Kong, J.; Wang, H.; Yang, C.; Jin, X.; Zuo, M.; Zhang, X. A spatial feature-enhanced attention neural network with high-order pooling representation for application in pest and disease recognition. *Agriculture* **2022**, *12*, 500. [\[CrossRef\]](#)
5. Li, X.; Li, M.; Yan, P.; Li, G.; Jiang, Y.; Luo, H.; Yin, S. Deep learning attention mechanism in medical image analysis: Basics and beyond. *Int. J. Netw. Dyn. Intell.* **2023**, *2*, 100006. [\[CrossRef\]](#)
6. Guarín-Torres, Y.C.; Patiño-Pacheco, M.J.; Martínez, J.W. Identificación del agente causal de la pudrición parda en frutos de duraznero (*Prunus persica*, L. Batsch) en Boyacá. *Entramado* **2019**, *15*, 298–309. [\[CrossRef\]](#)
7. Martini, C.; Mari, M. *Monilinia fructicola*, *Monilinia laxa* (Monilinia Rot, Brown Rot). In *Postharvest Decay: Control Strategies*; Bautista-Baños, S., Ed.; Elsevier: Amsterdam, The Netherlands, 2014; pp. 233–265. [\[CrossRef\]](#)
8. Hessane, A.; Youssefi, A.E.; Farhaoui, Y.; Aghoutane, B.; Amounas, F. A machine learning-based framework for stage-wise classification of date palm white scale disease. *Big Data Min. Anal.* **2023**, *6*, 263–272. [\[CrossRef\]](#)
9. Ministerio de Agricultura y Desarrollo Rural. Estadísticas Agronet 2022. Available online: <https://www.agronet.gov.co> (accessed on 10 January 2024).
10. Nobel, S.M.N.; Imran, M.A.; Bina, N.Z.; Kabir, M.M.; Safran, M.; Alfarhood, S.; Mridha, M.F. Palm leaf health management: A hybrid approach for automated disease detection and therapy enhancement. *IEEE Access* **2024**, *12*, 9097–9111. [\[CrossRef\]](#)
11. Septiarini, A.; Hamdani, H.; Hardianti, T.; Winarno, E.; Suyanto, S.; Irwansyah, E. Pixel quantification and color feature extraction on leaf images for oil palm disease identification. In *Proceedings of the 7th ICEEIE 2021*; IEEE: Piscataway, NJ, USA, 2021. [\[CrossRef\]](#)
12. Shruthi, U.; Nagaveni, V.; Raghavendra, B.K. A review on machine learning classification techniques for plant disease detection. In *Proceedings of the ICACCS 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 281–284. [\[CrossRef\]](#)
13. Mondal, J.J.; Islam, M.F.; Zabeen, S.; Islam, A.B.M.A.A.; Noor, J. PIEAD-Net: Identifying plant leaf diseases through leveraging limited-resource deep CNN. In *Proceedings of the ACM ICPC*; ACM: New York, NY, USA, 2022; pp. 668–673. [\[CrossRef\]](#)
14. Dolatabadian, A.; Neik, T.X.; Danilevich, M.F.; Upadhyaya, S.R.; Batley, J.; Edwards, D. Image-based crop disease detection using machine learning. *Plant Pathol.* **2025**, *74*, 18–38. [\[CrossRef\]](#)
15. Jafar, A.; Bibi, N.; Naqvi, R.A.; Sadeghi-Niaraki, A.; Jeong, D. Revolutionizing agriculture with artificial intelligence: Plant disease detection methods, applications, and their limitations. *Front. Plant Sci.* **2024**, *15*, 1356260. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Kaya, Y.; Gürsoy, E. A review of deep learning architectures for plant disease detection. *Turk. J. Biol.* **2025**, *49*, 459–497. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Mendes, J.; Lima, J.; Costa, L.; Rodrigues, N.; Pereira, A.I. Deep learning networks for olive cultivar identification. *Smart Agric. Technol.* **2024**, *8*, 100470. [\[CrossRef\]](#)
18. Talukder, M.S.H.; Sarkar, A.K. Nutrients deficiency diagnosis of rice crop by weighted average ensemble learning. *Smart Agric. Technol.* **2023**, *4*, 100155. [\[CrossRef\]](#)
19. Almannae, S.; Shatnawi, S.; Alshehhi, M. Date palm leaves discoloration detection system using deep transfer learning. In *Proceedings of the ETESIC 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 45–50.
20. Sharma, G.; Chaudhary, V.K.; Ahmad, D.; Shafeeq, M. Potato leaf disease detection using machine learning and deep learning techniques. In *Proceedings of the ARIIA 2024*; IEEE: Piscataway, NJ, USA, 2024. [\[CrossRef\]](#)
21. Samin, O.B.; Omar, M.; Mansoor, M. CapPlant: A capsule network based framework for plant disease classification. *PeerJ Comput. Sci.* **2021**, *7*, e752. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Shinoda, R.; Kataoka, H.; Hara, K.; Noguchi, R. Transformer-based ripeness segmentation for tomatoes. *Smart Agric. Technol.* **2023**, *4*, 100196. [\[CrossRef\]](#)
23. Ashurov, A.Y.; Al-Gaashani, M.S.A.M.; Samee, N.A.; Alkanhel, R.; Atteia, G.; Abdallah, H.A.; Muthanna, M.S.A. Enhancing plant disease detection through deep learning: A depthwise CNN with squeeze and excitation integration and residual skip connections. *Front. Plant Sci.* **2025**, *15*, 1505857. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Saeed, F.; Khan, M.A.; Sharif, M.; Mittal, M.; Goyal, L.M.; Roy, S. Deep neural network features fusion and selection based on PLS regression for crops diseases classification. *Appl. Soft Comput.* **2021**, *103*, 107164. [\[CrossRef\]](#)
25. Roy, K.; Chaudhuri, S.S.; Frnda, J.; Bandopadhyay, S.; Ray, I.J.; Banerjee, S.; Nedoma, J. Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet. *IEEE Access* **2023**, *11*, 14983–15001. [\[CrossRef\]](#)
26. Ashqar, B.A.M.; Abu-Naser, S.S. Image-based tomato leaves diseases detection using deep learning. *Int. J. Acad. Inf. Syst. Res.* **2019**, *3*, 10–16.
27. Yildiz, M.B.; Kursun, R.; Hafif, M.F.; Koksoy, E.K. Classification of diseases in tomato leaves using deep learning methods. *Intell. Methods Eng. Sci.* **2024**, *3*, 22–36. [\[CrossRef\]](#)
28. Mhala, P.; Bilandani, A.; Sharma, S. Enhancing crop productivity with fine-tuned deep CNN for potato leaf disease detection. *Expert Syst. Appl.* **2025**, *267*, 126066. [\[CrossRef\]](#)

29. Esgario, J.G.M.; Krohling, R.A.; Ventura, J.A. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Comput. Electron. Agric.* **2020**, *169*, 105162. [\[CrossRef\]](#)
30. Kim, W.S.; Lee, D.H.; Kim, Y.J. Machine vision-based automatic disease symptom detection of onion downy mildew. *Comput. Electron. Agric.* **2020**, *168*, 105099. [\[CrossRef\]](#)
31. Sambasivam, G.; Opiyo, G.D. A predictive machine learning application in agriculture: Cassava disease detection with imbalanced dataset using CNNs. *Egypt. Inform. J.* **2021**, *22*, 27–34. [\[CrossRef\]](#)
32. Hussain, N.; Khan, M.A.; Tariq, U.; Kadry, S.; Yar, M.A.E.; Mostafa, A.M.; Alnuaim, A.A.; Ahmad, S. Multiclass cucumber leaf diseases recognition using best feature selection. *Comput. Mater. Contin.* **2022**, *70*, 3281–3294. [\[CrossRef\]](#)
33. Abdel-Basset, M.; Kunduracioglu, I.; Pacal, I. Utilizing deep learning models for early detection and classification of fruit diseases. *Sci. Rep.* **2026**, *16*, 8167. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Khan, M.A.; Akram, T.; Sharif, M.; Awais, M.; Javed, K.; Ali, H.; Saba, T. CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on CNN features. *Comput. Electron. Agric.* **2018**, *155*, 220–236. [\[CrossRef\]](#)
35. Bedda, H. Apple Leaf Diseases Detection Using Deep Learning. Master's Thesis, University of Banta 2 (Mostafa Ben Bouliad), Fesdis, Algeria, 2023. [\[CrossRef\]](#)
36. Elngar, A.A.; Arafa, M.; Fathy, A.; Moustafa, B.; Mahmoud, O.; Shaban, M.; Fawzy, N. Image classification based on CNN: A survey. *J. Cybersecur. Inf. Manag.* **2021**, *6*, 18. [\[CrossRef\]](#)
37. Nababan, M.; Laia, Y.; Sitanggang, D.; Sihombing, O.; Indra, E.; Siregar, S.; Purba, W.; Mancur, R. The diagnose of oil palm disease using naive Bayes method based on expert system technology. *J. Phys. Conf. Ser.* **2018**, *1007*, 012015. [\[CrossRef\]](#)
38. Asraf, H.M.; Nooritawati, M.T.; Rizam, M.S.B.S. A comparative study in kernel-based SVM of oil palm leaves nutrient disease. *Procedia Eng.* **2012**, *41*, 1353–1359. [\[CrossRef\]](#)
39. Ong, J.H.; Ong, P.; Lee, W.K. Image-based oil palm leaf disease detection using CNN. *J. Inf. Commun. Technol.* **2022**, *21*, 383–410. [\[CrossRef\]](#)
40. Aji, A.F.; Munajat, Q.; Pratama, A.P.; Kalamullah, H.; Aprinaldi; Setiyawan, J.; Arymurthy, A.M. Detection of palm oil leaf disease with image processing and neural network on mobile device. *Int. J. Comput. Theory Eng.* **2013**, *5*, 528–532. [\[CrossRef\]](#)
41. Hessane, A.; Youssefi, A.E.; Farhaoui, Y.; Aghoutane, B. Toward a stage-wise classification of date palm white scale disease using ML techniques. In *Proceedings of the ISCV 2022*; IEEE: Piscataway, NJ, USA, 2022. [\[CrossRef\]](#)
42. Al-Mahmood, A.M.; Shahadi, H.I.; Khayeat, A.R.H. Image dataset of infected date palm leaves by dubas insects. In *Proceedings of the ICECET 2022*; IEEE: Piscataway, NJ, USA, 2022. [\[CrossRef\]](#)
43. Liaghat, S.; Ehsani, R.; Mansor, S.; Shafri, H.Z.M.; Meon, S.; Sankaran, S.; Azam, S.H.M.N. Early detection of basal stem rot disease (Ganoderma) in oil palms based on hyperspectral reflectance data. *Int. J. Remote Sens.* **2014**, *35*, 3427–3439. [\[CrossRef\]](#)
44. Bargoti, S.; Underwood, J. Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* **2017**, *34*, 1039–1060. [\[CrossRef\]](#)
45. Pal, C.; Karmakar, S.; Mukherjee, I.; Chakrabarti, P.P. A lightweight and explainable CNN model for empowering plant disease diagnosis. *Sci. Rep.* **2025**, *15*, 30720. [\[CrossRef\]](#) [\[PubMed\]](#)
46. Jelali, M. Deep learning networks-based tomato disease and pest detection: A first review using real field datasets. *Front. Plant Sci.* **2024**, *15*, 1493322. [\[CrossRef\]](#) [\[PubMed\]](#)
47. Polder, G.; Blok, P.M.; van Daalen, T.; Peller, J.; Mylonas, N. A smart camera with integrated deep learning processing for disease detection in open field crops. *J. Field Robot.* **2025**, *42*, 2062–2075. [\[CrossRef\]](#)
48. Sodjinou, S.G.; Mohammadi, V.; Mahama, A.T.S.; Gouton, P. A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images. *Inf. Process. Agric.* **2022**, *9*, 355–364. [\[CrossRef\]](#)
49. Cheng, X.; Zhang, Y.; Chen, Y.; Wu, Y.; Yue, Y. Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* **2017**, *141*, 351–356. [\[CrossRef\]](#)
50. Faisal, M.; Albogamy, F.; Elgibreen, H.; Algabri, M.; Alqershi, F.A. Deep learning and computer vision for estimating date fruits type, maturity level, and weight. *IEEE Access* **2020**, *8*, 206770–206782. [\[CrossRef\]](#)
51. Dias, P.A.; Tabb, A.; Medeiros, H. Apple flower detection using deep convolutional networks. *Comput. Ind.* **2018**, *99*, 17–28. [\[CrossRef\]](#)
52. Asad, M.H.; Bais, A. Weed detection in canola fields using maximum likelihood classification and deep CNN. *Inf. Process. Agric.* **2020**, *7*, 535–545. [\[CrossRef\]](#)
53. Barth, R.; IJsselmuiden, J.; Hemming, J.; Henten, E.J.V. Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Comput. Electron. Agric.* **2019**, *161*, 291–304. [\[CrossRef\]](#)
54. Wang, F.; Wang, R.; Xie, C.; Yang, P.; Liu, L. Fusing multi-scale context-aware information for automatic in-field pest detection and recognition. *Comput. Electron. Agric.* **2020**, *169*, 105222. [\[CrossRef\]](#)
55. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360. [\[CrossRef\]](#)

56. Selvaraju, R.R.; 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 (ICCV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 618–626. [[CrossRef](#)]
57. Fan, Y.; Yu, M.; Shen, L.; Ma, J.; Zeng, Z.; Wang, H. Robust plant disease segmentation in complex field environments: STAR-Net. *Front. Plant Sci.* **2026**, *16*, 1706072. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.