AgriEngineering

Article

Lightweight Hybrid Deep Learning for Strawberry Disease Recognition and Edge Deployment Using Dynamic Multi-Scale CNN–Transformer Fusion

Nasreddine Haqiq [1,*], Mounia Zaim [1], Mohamed Sbihi [1], Mustapha El Alaoui [2], Khalid El Amraoui [2], Youssef El Kazini [2], Hassane Roukhe [2] and Lhoussaine Masmoudi [2]

[1] Laboratory of Systems Analysis, Information Processing and Industrial Management (LASTIMI), High School of Technology of Salé, Mohammed V University, Rabat 10000, Morocco; mounia.zaim@est.um5.ac.ma (M.Z.); mohammed.sbihi@est.um5.ac.ma (M.S.)

[2] LCS Laboratory, Physics Department, Faculty of Sciences, Mohammed V University, Rabat 10000, Morocco; m.elalaoui@um5r.ac.ma (M.E.A.); khalid.elamraoui@um5r.ac.ma (K.E.A.); youssef.elkazini@um5r.ac.ma (Y.E.K.); hassane.roukhe@fsr.um5.ac.ma (H.R.); l.masmoudi@um5r.ac.ma (L.M.)

* Correspondence: nasreddine_haqiq@um5.ac.ma; Tel.: +212-629175557

Abstract

To implement a successful strawberry (Fragaria × ananassa) farming, fungal diseases must be detected in a timely manner so that informed crop protection decisions can be made. While field scouting is an option, it is manual and labor intensive. Scouting is also inaccurate and reduces efficiency due to micro-climatic lighting and field clutter, among other factors. StrawberryDualNet is a framework that supports Integrated Pest Management and automates symptom surveillance. We present dual-path CNN–Transformer fusion design that integrates two branches: a dynamic multi-scale convolution and a lightweight transformer. The former is able to capture fine-grained morphological lesion textures, while the latter captures overall contextual patterns. The two representations are fused through a learnable gating mechanism to decrease visual uncertainty amongst differing symptoms. We used a stratified five-fold cross-validation to evaluate the framework among five economically significant pathogens. Our approach significantly outperformed other automated scouting baselines, achieving 95.1% accuracy and 95.3% precision, respectively, and it is successful for Anthracnose, Gray Mold, Powdery Mildew, Rhizopus Rot, and Black Spot. The model is also scaled down compared to others (0.04 M parameters; 0.72 MB, 13–20× smaller than MobileNetV2/ShuffleNetV2) and is thus able to be deployed on devices that are lacking computational resources. For edge feasibility, we assessed reduced-precision inference; 16-bit floating point quantization preserved baseline performance at 83 FPS, whereas 8-bit integer quantization caused notable accuracy degradation. Overall, the proposed local–global fusion design provides an accurate, interpretable, and scalable tool for real-time disease phenotyping in precision horticulture.

Keywords: strawberry disease detection; deep learning; hybrid CNN–Transformer; image processing; precision agriculture

Check for updates

Academic Editors: Ray E. Sheriff and Chiew Foong Kwong

Received: 16 January 2026
Revised: 12 February 2026
Accepted: 18 February 2026
Published: 22 February 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) license.

1. Introduction

Strawberry production is strongly affected by fungal and bacterial diseases, which can reduce yield and market quality if symptoms are not detected early [1]. In real fields, symptoms are often subtle and diverse. They can also change with lighting, background

clutter, occlusions, cultivar differences, and growth stage. These factors make routine visual diagnosis unreliable and can delay crop protection decisions. For this reason, image-based disease assessment is becoming an important part of precision agriculture, because it can support scalable and consistent monitoring of plant health.

Traditional disease scouting uses manual visual inspection, which is labor-intensive and prone to observer subjectivity and fatigue [2]. Meanwhile, laboratory-based diagnostic protocols cannot support continuous monitoring [3]. Early computer vision work used handcrafted color and texture descriptors, but these methods were sensitive to scale, orientation, and environmental changes; later, CNNs became popular because they learn features directly from images [4]. However, CNNs mainly capture local patterns and often miss long-range relationships, which help tell similar symptoms apart in complex backgrounds. Transformer-based models use self-attention to capture global dependencies. However, these models often miss fine texture details, which are essential for telling diseases apart that look similar at first glance. Recent work shows that combining local details with global structure improves fine-grained recognition [5]. Similar findings apply to plant disease recognition using different imaging methods, including hyperspectral imaging [6]. These findings motivated hybrid CNN–Transformer models. These models combine local texture and global context for robust diagnosis in real agricultural images [7]. In practice, deployment remains challenging on edge devices, and it often needs heavy optimization to run efficiently [8]. Most existing frameworks only classify diseases. They do not locate lesions. Yet location information is important for agricultural decisions. Recent work using instance segmentation shows that locating disease regions improves decision reliability [9]. Therefore, we need a compact framework that both classifies diseases and locates lesions for practical strawberry monitoring.

Hybrid CNN–Transformer models have been used for plant disease detection, but existing approaches use fixed architectures that cannot adapt to variable lesion scales. To address these challenges, we propose StrawberryDualNet, a dual-path Convolutional Neural Network–Transformer deep learning architecture for strawberry disease identification and diseased-region localization. The model comprises two cooperative branches: (i) a CNN-based local branch based on a Dynamic Multi-Scale Convolution Module (DMSCM) that captures fine texture and color variations using parallel depthwise separable convolutions with different receptive fields; and (ii) a Transformer-based global branch that converts intermediate feature maps into tokens processed by a lightweight Transformer encoder to model long-range dependencies and contextual semantics. A learnable FusionGate adaptively integrates local and global representations to promote feature complementarity and reduce redundancy. To improve stability under small and imbalanced data, we further adopt a best-split K-Fold strategy with multi-seed training, following evidence supporting optimized cross-validation protocols [10], and refer to this setting as StrawberryDualKFold. Finally, disease-specific image processing is applied to highlight affected regions, providing interpretable outputs that complement the predicted disease class.

To position our contribution against recent lightweight hybrid architectures, we explicitly contrast StrawberryDualNet with state-of-the-art deployment-oriented models. MobileViT-DAP [11] uses MobileViT-XXS with attention and lightweight mixing for rice disease classification, applying fixed receptive-field designs and generic late fusion without dynamic scale selection. TinyResViT [12] combines ResNet with vision-transformer reasoning for corn leaf disease detection, designed for embedded hardware such as Raspberry Pi 4 and Jetson Nano with simple concatenation-based fusion, but lacks input-dependent gating mechanisms. StrawberryDualNet introduces input-adaptive multi-scale weighting through DMSCM—an MLP that dynamically selects optimal receptive fields (3×3, 5×5, 7×7) based on per-image lesion characteristics rather than fixed backbone weights. Our Fusion-

Gate uses learnable sigmoid gating with residual connections across multiple resolutions, preventing gradient degradation while balancing local texture and global context. Neither MobileViT-DAP nor TinyResViT addresses fruit-level lesion variability with hierarchical adaptive fusion: their fixed architectures cannot adapt to the scale diversity of strawberry diseases ranging from pinhead-sized black spots to diffuse powdery mildew coverage. These architectural distinctions—dynamic kernel selection and learnable cross-resolution gating—constitute the core methodological novelty of StrawberryDualNet, allowing lesion-discriminative feature learning not achieved in prior lightweight hybrid architectures.

The main contributions of this work are summarized as follows: (1) We present a dual-path (CNN–Transformer) framework for strawberry disease diagnosis with dynamic multi-scale selection and learnable fusion gates. Unlike existing lightweight hybrids with fixed architectures, our model adapts receptive fields and feature weighting to input-specific lesion characteristics and integrates CNN-based local texture modeling and Transformed-based global contextual reasoning within a compact architecture suitable for data-limited agricultural settings. (2) We introduce the DMSCM and FusionGate modules for adaptive multi-scale feature fusion at low computational cost. (3) We propose StrawberryDualKFold, a training protocol that reduces sensitivity to data partitioning. It uses best-split K-Fold selection and multi-seed aggregation. (4) We demonstrate improved performance over CNN and Transformer baselines on five strawberry diseases. We also provide lesion localization maps to support field decisions.

Overall, this work presents a practical deep learning system for automated strawberry disease diagnosis. It shows how combining local and global visual cues supports precision agriculture.

2. Related Work

Automated strawberry disease recognition is important in precision agriculture because early diagnosis helps farmers react faster and reduce yield and quality losses under field conditions [13]. However, available strawberry datasets are often small and imbalanced, and many studies do not report stability across different splits or random seeds, which makes real-world performance less certain. Existing methods mainly include (i) traditional vision or spectral sensing, (ii) CNN-based classification/detection/segmentation, and (iii) Transformer or hybrid CNN–Transformer models. This section reviews these directions and summarizes the key gaps for practical strawberry monitoring.

2.1. Strawberry Disease Recognition with CNNs and Lesion Localization

Early studies used classical vision pipelines or hyperspectral imaging with handcrafted spectral/texture features for early detection under controlled settings [14]. While useful, these hand-designed features can be sensitive to changes in cultivar, growth stage, and illumination. CNNs are now widely used for strawberry disease classification and often achieve strong accuracy with simple training setups [15]. For complex backgrounds and small lesions, detector-based methods and YOLO variants have been adapted to field images to improve small-object detection [16,17]. For localization, instance segmentation (e.g., Mask R-CNN) can produce lesion masks but needs pixel-level labels and is hard to scale when expert annotation is limited [18]. Weaker-supervision options, such as class-attention lesion proposals, can also highlight disease regions with less labeling effort [19]. Still, many CNN pipelines struggle to use global context, which matters when diseases look similar locally or when background affects appearance.

2.2. Transformer-Based Global Reasoning and Hybrid Local–Global Fusion

Transformers model long-range relations using self-attention and can reduce confusion when local symptoms are similar by using broader context. In plant disease tasks, multi-branch Transformer designs with deep supervision have improved robustness and severity-related learning [20]. In strawberry disease recognition, attention-enhanced modules have been used to improve class separation in cluttered scenes by combining convolution and attention [21]. Hybrid CNN–Transformer models are a common compromise: CNN parts capture fine lesion details, while Transformer parts capture wider context, and they have shown gains in crop pest and disease identification [22]. For deployment, efficiency is also important; lightweight models such as StrawberryNet aim to balance accuracy and real-time use [23]. However, many works focus on classification only or handle localization as a separate step, which can cause a mismatch between the predicted class and the visual evidence.

Recent work increasingly targets lightweight models that keep good accuracy while remaining practical for edge deployment. MobileViT-DAP [11] is a small CNN–Transformer model for rice disease classification (0.75 M parameters, 0.23 GFLOPs, 3.03 MB) with 5.15 ms latency on CPU and real-time speed on GPU. TinyResViT [12] combines ResNet with Transformer modules for corn leaf disease detection and reports 52.67 FPS on Raspberry Pi 4 and Jetson Nano (1.59 GFLOPs). Gookyi et al. [24] used Edge Impulse to deploy several CNN backbones (MobileNet, EfficientNet, ShuffleNet, SqueezeNet) with TFLite INT8 quantization for tomato disease detection, reaching 97.12% accuracy with EfficientNet (4.60 MB). I-GhostNetV3 [25] adds attention to GhostNetV3 for rice and reports 1.831 M parameters and 248.694 MFLOPs for vision-sensor-based smart agriculture use. Light-MobileBerryNet [26] targets strawberry disease recognition with MobileNetV3 (0.53 M parameters, 2 MB) and provides Grad-CAM visual explanations, with 96.6% accuracy on mobile deployment. Overall, these methods show strong deployment results, but they mainly use fixed backbones and fixed fusion designs, without input-adaptive multi-scale selection or learnable gating. Table 1 summarizes their supervision type, complexity, and deployment settings.

Table 1. Comparative summary of recent lightweight methods for plant disease recognition.

Study	Model	Crop	Supervision	Model Size/Complexity	Deployment Analysis
Gookyi et al. (2024) [24]	CNN	Tomato	Image-level supervised	4.60 MB TFLite INT8	Edge Impulse platform edge device deployment
Zhang et al. (2025) [11]	MobileViT-DAP (Hybrid)	Rice	Supervised (hard labels)	0.75 M params 3.03 MB 0.23 GFLOPs	5.15 ms latency (CPU) real-time GPU
Truong-Dang et al. (2025) [12]	TinyResViT (Hybrid)	Corn	Supervised (cross-entropy)	1.59 GFLOPs lightweight architecture	52.67 FPS (Raspberry Pi 4) Jetson Nano
I-GhostNetV3 (2026) [25]	GhostNetV3 + Attention	Rice	Supervised with attention	1.831 M params 248.694 MFLOPs	Vision-sensor-based smart agriculture deployment
Light-MobileBerryNet (2025) [26]	MobileNetV3 + Grad-CAM	Strawberry	Interpretable supervised	0.53M params 2 MB	Mobile deployment 96.6% accuracy Grad-CAM visualization

2.3. Objectives

The aim of this study is to build and validate a compact deep learning framework for strawberry disease recognition and visual symptom evidence, under practical limits such as small and imbalanced datasets and edge deployment constraints.

The objectives are to: (i) develop a lightweight hybrid model that uses both local lesion texture and global context for multi-class disease classification; (ii) test robustness on limited and imbalanced data using repeated seeds and a stability-based split protocol (StrawberryDualKFold); (iii) provide symptom-region evidence using simple post-processing adapted to each disease; and (iv) assess deployment readiness via TensorFlow Lite export and reduced-precision inference.

Our main assumption is that combining dynamic multi-scale convolution with transformer-based context, together with stability-aware evaluation, improves performance over standard CNN baselines while staying suitable for resource-limited devices.

3. Materials and Methods

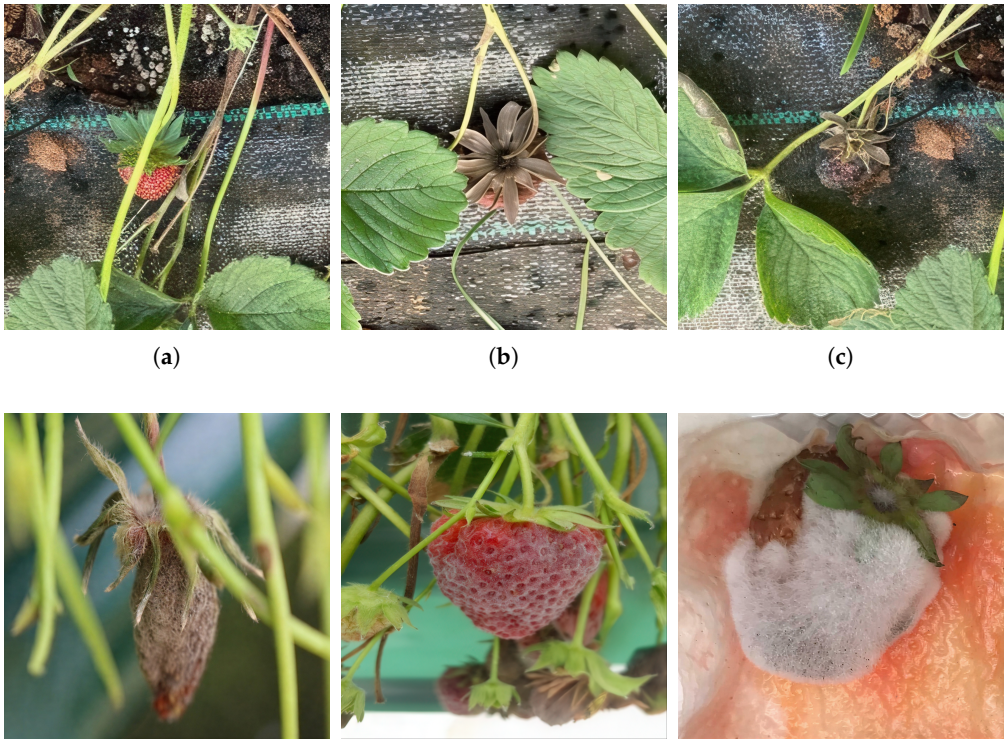
This section presents the dataset and disease categories, followed by the proposed StrawberryDualNet framework for strawberry disease recognition and lesion localization. The method description is provided with sufficient detail to enable replication.

3.1. Image Dataset

The data used in these experiments were derived from the publicly accessible Strawberry Disease Detection Dataset ("Instance Segmentation Dataset for Seven Types of Strawberry Diseases"), accessible from the JBNU (Jeonbuk National University) Artificial Intelligence Laboratory [27]. This dataset contains 2500 segmented strawberry images, each annotated with a pixel-level segmentation mask for one of the seven strawberry disease categories. The dataset is a combination of field and greenhouse images in natural lighting and is supplemented with images from freely accessible agricultural datasets. Thus, the dataset has high variability in background clutter, lighting, and distance of the camera to the subject (close-up to wide angle) and in the different stages of disease manifestation. This type of dataset variability is important for testing the robustness of models in authentic field scouting conditions. Figure 1 shows representative images used in our experiments, including samples from the JBNU dataset.

We applied simple quantitative rules to curate the image set and remove unclear samples. First, we kept an image only if a strawberry fruit is clearly visible: after extracting a coarse fruit mask, the fruit region had to cover at least 10% of the image area; otherwise the image was excluded (fruit too far, mostly leaf/background, or no fruit). When multiple fruits appeared, we kept the image only if the largest fruit region accounted for at least 70% of all fruit pixels to keep a clear single target. Second, we removed low-visibility images using basic quality checks: blurred images were detected using the variance of Laplacian on the grayscale image, and samples with Laplacian variance below a fixed threshold ($T_{\text{blur}} \approx 40$–$80$ on resized 224×224 images) were excluded; we also removed extreme illumination cases using HSV V-channel statistics, excluding images if $\text{mean}(V) < 0.20$ (too dark) or $\text{mean}(V) > 0.90$ (too bright), or if more than 15% of pixels were near-black ($V < 0.05$) or near-white ($V > 0.95$). Third, we filtered heavy occlusion or partial-fruit cases using geometry-based criteria: images were excluded when the fruit mask touched the image border (within 2–3 pixels after resizing), indicating cropping, or when the fruit shape was too fragmented, measured by low solidity $\left(\text{Area}(\text{mask})/\text{Area}(\text{convex hull}) < 0.80\right)$ or low fill ratio $\left(\text{Area}(\text{mask})/\text{Area}(\text{bounding box}) < 0.55\right)$. These quantitative rules reduce label noise and make the dataset curation more reproducible. From the 2500 source images, 596 were excluded by the curation rules. These exclusions mainly include non-fruit/leaf-

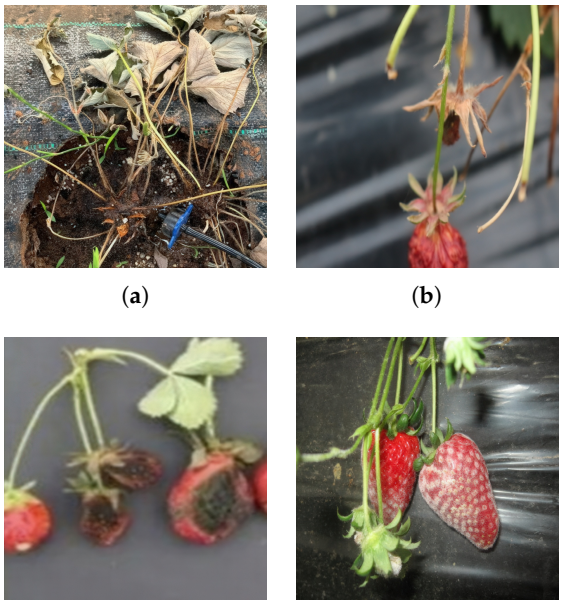
only samples and low-quality cases (strong occlusion, blur, or extreme lighting). The excluded images were distributed as follows: non-fruit/leaf-only (340), Healthy (121), Anthracnose (57), Gray Mold (45), and Powdery Mildew (38). No samples were removed from the Black Spot and Rhizopus Rot categories. Example excluded cases are provided in Figure 2.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 1. Representative strawberry image samples used in this study. (**a**) Healthy; (**b**) Healthy (flower stage); (**c**) Black Spot; (**d**) Gray Mold; (**e**) Powdery Mildew; (**f**) Rhizopus Rot.



(**a**) (**b**)

(**c**) (**d**)

Figure 2. Excluded samples and triggered rules: (**a**) no fruit ($r < 0.10$); (**b**) blur ($s < T_{\text{blur}}$); (**c**) small + blur ($r < 0.10$, $s < T_{\text{blur}}$); (**d**) no dominant target/occluded fruit ($r_{\text{dom}} < 0.70$ and/or solidity < 0.80 or fill ratio < 0.55).

To improve model robustness and limit overfitting, we applied light data augmentation during training. This step is important because strawberry disease datasets are usually limited in size, and deep models can otherwise learn background or acquisition-specific cues (e.g., lighting, camera angle, leaf positions) instead of true disease symptoms. First, all images were resized to 128 × 128 and converted to RGB. Then, for each training epoch, every image could be randomly transformed to reproduce realistic changes that happen during image capture, such as small viewpoint variations, minor camera movements, and illumination differences. The applied transformations were random horizontal flip, random rotation within ±20°, random horizontal and vertical shifts up to ±15% of the image size, random zoom in/out within 0.85–1.15, and random brightness scaling within 0.8–1.2. For geometric operations, any empty areas created at the borders were filled using nearest-neighbor padding. These augmentations increase the diversity of the training data without changing the disease label, helping the model generalize better to new images and reducing the risk of overfitting.To visually clarify these transformations, Figure 3 shows representative outputs of each augmentation applied to the same image.

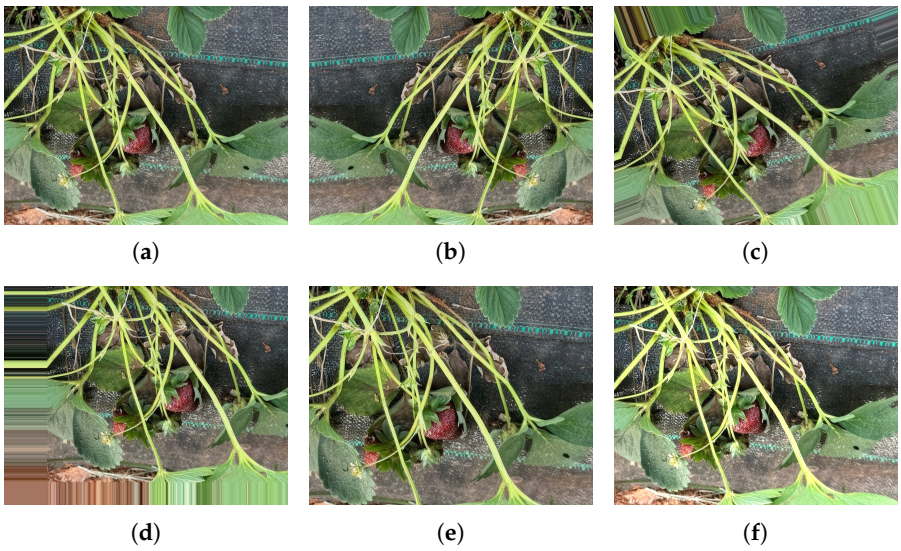


Figure 3. Examples of the light data augmentation applied during training: (**a**) original image; (**b**) horizontal flip; (**c**) rotation (within ±20°); (**d**) translation (up to ±15% along both axes); (**e**) zoom in/out (0.85–1.15); (**f**) brightness scaling (0.8–1.2).

We prepare the curated strawberry dataset by resizing all images to 128 × 128, converting them to RGB, and normalizing pixel values to $[0, 1]$. To reduce overfitting on a limited and imbalanced dataset, we apply light data augmentation during training: random horizontal flip, rotation within ±20°, width/height shifts up to ±15%, zoom 0.85–1.15, and brightness scaling 0.8–1.2. These transforms mimic common capture changes (camera angle, small motion, and lighting) without changing the disease label. For each original image, we generate three augmented variants, expanding the dataset from 1904 original images to 7616 images in total. Augmentation is applied to the training data only, while validation and testing use the original (non-augmented) images. Table 2 summarizes the curated dataset composition, reports the original counts and the expanded counts after augmentation (three variants per original) for each class.

Table 2. Original and augmented image counts per class (three augmented variants per original image).

Class	Original Data	Augmented Data	Total Data
Healthy	331	993	1324
Anthracnose	240	720	960
Gray Mold	365	1095	1460
Powdery Mildew	220	660	880
Rhizopus Rot	348	1044	1392
Black Spot	400	1200	1600
All	1904	5712	7616

3.2. Disease Categories and Annotation Protocol

The classification task in this work targets the automatic recognition of five major fungal diseases affecting strawberry production. Each disease class was chosen according to its agronomic importance and the presence of distinct visible symptoms in RGB images. Annotations of each disease were made under the direction of experts in order to ensure that typical lesion patterns and also variability in healthy tissue were able to be captured through different viewing conditions.

The five diseases are very well defined: Gray Mold, Anthracnose, Powdery Mildew, Rhizopus Rot, and Black Spot, each of which manifests differently on the fruit and foliage. Gray Mold asserts itself as grayish tawny fuzzy mold on ripe berries and blossoms in humid conditions. Anthracnose is characterized by dark sunken lesions on fruits and leaves and can cause severe fruit rot and wilting of foliage. Powdery Mildew causes white powdery patches to form on the leaf surface and, in cases untreated with fungicide, induces leaf curling and chlorosis. Rhizopus Rot (Rhizopus stolonifer) appears as rapidly progressing, soft and watery decay on harvested berries, commonly overgrown with coarse, gray mycelia. Finally, Black Spot is characterized by small, dark lesions on leaves or stems that primarily degrade visual quality but may also reduce overall plant vigor. The relative severity and importance of these diseases are summarized in Table 3.

Table 3. Detailed overview of strawberry diseases, indicative symptoms, and criticality ranking.

Disease	Identification Features	Criticality
Gray Mold	Grayish-brown, fuzzy fungal growth on ripe fruits and flowers; spreads quickly in damp conditions, causing significant pre- and post-harvest losses.	4
Powdery Mildew	White, powdery fungal patches on leaf surfaces, undersides, and stems; may cause curling and chlorosis and reduce photosynthesis.	2
Anthracnose	Dark, sunken lesions on fruits and leaves; can lead to severe fruit rot, leaf wilting, and significant yield losses if not controlled promptly.	5
Rhizopus Rot	Soft, watery decay on harvested/stored fruits, with coarse grayish fungal growth thriving under warm and humid postharvest conditions.	3
Black Spot	Small, circular to irregular dark lesions on leaves and occasionally stems; generally less severe but can contribute to leaf drop and reduced vigor.	1

3.3. Overview of the Proposed StrawberryDualNet Framework

The proposed StrawberryDualNet framework is designed for dual-stage strawberry disease recognition and lesion localization (Figure 4). The model integrates two complementary components: (1) a local feature extraction branch based on a Dynamic Multi-Scale Convolution Module (DMSCM), which captures fine texture and color patterns related to

strawberry surface abnormalities, and (2) a global reasoning branch using a lightweight Transformer encoder that models contextual relationships between regions. A learnable FusionGate adaptively merges the outputs of both branches, enabling the network to combine detailed local information with global semantics. The resulting fused features are used for classification and for generating interpretable lesion activation maps. The architecture is trained end-to-end using a best-split K-Fold strategy and a multi-seed protocol to improve robustness on small, imbalanced datasets [28].

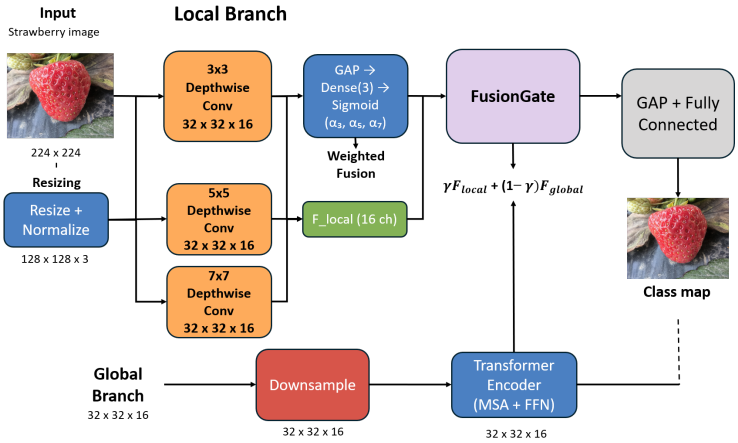


Figure 4. Framework of the proposed StrawberryDualNet model. It uses a dual-branch design. The dynamic multi-scale convolution (DMSCM) extracts local features, and the transformer encoders model global context. The arrows show the feature flow. The FusionGate module (purple) adaptively fuses local and global features using learned gating coefficients γ, which supports feature integration across multiple resolutions. Stage 1 operates at 32×32 with 16 channels, and Stage 2 operates at 16×16 with 32 channels.

3.4. Dual-Branch Feature Extraction

The feature extraction stage constructs a joint representation that preserves both fine local texture and global structural context. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, pixel intensity values are normalized to $[0, 1]$ and resized to a fixed resolution used by the network (here denoted (h, w)). The extraction process is realized in two branches that should cooperate. The first branch is local, and it is mainly intended to capture short-range correlations with a very basic focus. The second is global, and it captures long-range associations through a transformer.

3.4.1. Local Branch: Dynamic Multi-Scale Convolution Module (DMSCM)

The Dynamic Multi-Scale Convolution Module uses a parallel multi-branch architecture to efficiently capture discriminative micro-patterns that indicate early-stage infection, such as subtle color changes, mold textures, and delicate surface details. In practice, it executes three depthwise separable convolutions with various kernel sizes (3×3, 5×5, and 7×7) [29]. Within the DMSCM, each of the three parallel branches performs a separable depthwise convolution, each branch first applies a depthwise convolution that maintains the depth of the input channel, followed by a pointwise convolution (1×1) that projects the features into the C' output channels, where $C' = 16$ in the first stage and $C' = 32$ in the second stage, which produces three outputs of the branches $B_k \in \mathbb{R}^{H \times W \times C'}$ for $k \in \{1, 2, 3\}$.

The module presents a lightweight adaptive fusion mechanism that learns a distinct weight for each scale [30], as opposed to merging these multi-scale features with uniform weights. Through dynamic modulation of each kernel size's influence based on the input image's content, the network can select the most informative scale for every instance. This

adaptive behavior markedly enhances robustness to differences in lesion size and camera distance, both of which are frequent sources of variation in field-collected images.

Formally, let $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ denote the input feature map. The DMSCM applies $K = 3$ parallel depthwise separable convolution branches with kernel sizes $\mathcal{K} = \{3, 5, 7\}$, producing branch outputs $\{\mathbf{B}_k\}_{k=1}^{K}$ where each $\mathbf{B}_k \in \mathbb{R}^{H \times W \times C'}$.

The dynamic weighting mechanism operates as follows. First, global average pooling compresses spatial dimensions to obtain channel-wise statistics:

$$\mathbf{z} = \mathrm{GAP}(\mathbf{X}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X}_{i,j} \in \mathbb{R}^{C} \tag{1}$$

A learnable linear projection with sigmoid activation computes kernel-specific importance weights:

$$\boldsymbol{\alpha} = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b}) \in \mathbb{R}^{K} \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{C \times K}$ and $\mathbf{b} \in \mathbb{R}^{K}$ are learned parameters, and $\sigma(\cdot)$ denotes the sigmoid function ensuring $\alpha_k \in (0, 1)$.

The fused output combines branch outputs with learned weights:

$$\mathbf{Y} = \sum_{k=1}^{K} \alpha_k \mathbf{B}_k \tag{3}$$

This formulation enables input-dependent scale selection: for images with fine speck lesions, α_1 (3×3 kernel) dominates; for diffuse mildew coverage, α_3 (7×7 kernel) increases. The gating parameters are trained end-to-end via backpropagation through the classification loss, with no additional supervision required.

3.4.2. Global Branch: Lightweight Transformer Encoder

Although convolutional layers successfully understand small details, they aren't always good at understanding larger contexts that differentiate similar symptoms. The global branch bypasses this limitation by using a lightweight transformer encoder to understand long-range dependencies on the entire surface of the fruit [31]. The model transforms intermediate feature maps into a series of patch tokens to be assessed by a multi-head self-attention module.

Thus, the model can think about and reason the relationships of different, possibly distant regions (for instance, a dispersed spot and a localized patch of a decay area) without a limited receptive field. Such global context modeling is especially important for separating diseases that share similar local texture patterns but exhibit different spatial arrangements [32]. Finally, the outputs from both the local and global branches are normalized to maintain balanced feature scales before entering the fusion stage.

3.5. Network Architecture

The complete network architecture is illustrated in Figure 5. It consists of three primary stages: (1) dual-branch feature extraction (DMSCM + Transformer Encoder), (2) hierarchical feature fusion via FusionGate, and (3) final classification. The upper pathway extracts multi-scale local features using depthwise separable convolutions (3×3, 5×5, and 7×7). The lower pathway performs global reasoning by dividing the input into patches and processing them with a lightweight Transformer encoder. Outputs from both branches are fused at multiple resolutions through FusionGate, which learns channel-wise importance weights to adaptively combine local and global descriptors [33]. The fused representation

is aggregated by global average pooling and projected through two fully connected layers ($32 \rightarrow 5$) to produce class probabilities.

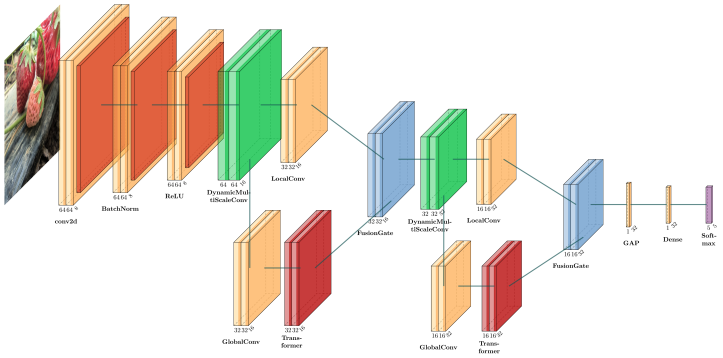


Figure 5. Detailed architecture of the StrawberryDualNet backbone. The arrows show the data flow between the main modules. The Dynamic Multi-Scale Convolution Module (DMSCM, orange) uses learned weights α to combine receptive fields (3×3, 5×5, 7×7). The FusionGate modules (purple) use gating factors γ to blend local and global features. Independent parameters are learned at each resolution (Stage 1: 16 channels; Stage 2: 32 channels). The transformer encoders (red) model long-range context using multi-head self-attention (MSA) and feed-forward networks (FFN). Numbers indicate feature map size (height $\times$ width $\times$ channels).

3.6. Training Objective

The network is trained end-to-end using a supervised learning objective. For each image I_i, the model outputs a class-probability vector $\hat{\mathbf{y}}_i \in \mathbb{R}^C$, where $C = 5$, and the ground-truth label is represented as a one-hot vector $\mathbf{y}_i$. The categorical cross-entropy loss [34] is used:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}). \tag{4}$$

To reduce overfitting, dropout is applied before the fully connected layers and weight decay is incorporated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \|\theta\|_2^2, \tag{5}$$

where λ controls the L_2 regularization strength and θ denotes model parameters.

3.7. Validation Protocol and Best-Split Selection

Standard K-Fold cross-validation averages performance across all folds. This hides variability from individual split quality. We propose StrawberryDualKFold, a two-stage protocol. First, we select the single most representative fold based on minimum class distribution deviation. Second, we repeat training with multiple random seeds on this fixed split. This isolates split quality from optimization stability, providing both a reproducible benchmark and reliable variance estimates. For fold j, the deviation metric is defined as:

$$M_j = \sum_{c=1}^{C} \left| r_{j,\text{train}}^c - r^c \right| + \sum_{c=1}^{C} \left| r_{j,\text{test}}^c - r^c \right|, \tag{6}$$

where r^c is the overall fraction of samples in class c, and $r_{j,\text{train}}^c$ and $r_{j,\text{test}}^c$ are the corresponding fractions in the training and test subsets for fold j. The fold minimizing M_j is selected as the canonical split. To verify that the best-split selection does not introduce optimistic bias, we conducted a 5-fold cross-validation using identical training configurations for all folds. The results are consistent across folds (accuracy 0.9324 ± 0.0143) in Table 4, and the canonical Fold 3 (0.9328) falls within one standard deviation of the mean. This indicates

that selecting the minimum-deviation fold yields a statistically representative split rather than an overly favorable one.

Table 4. Fold-wise distribution deviation and cross-validation performance metrics in the 5-fold scheme. Fold 3, with the lowest deviation score, is selected as the canonical split (StrawberryDualKFold).

Fold	Deviation	Accuracy	Precision	Recall	F1
0	0.0867	0.9451	0.9427	0.9461	0.9440
1	0.1216	0.9348	0.9379	0.9340	0.9348
2	0.1116	0.9084	0.9111	0.9161	0.9132
3	0.0637	0.9328	0.9330	0.9335	0.9323
4	0.0800	0.9409	0.9405	0.9445	0.9423
Mean ± Std	0.0927 ± 0.0210	0.9324 ± 0.0143	0.9330 ± 0.0128	0.9348 ± 0.0120	0.9333 ± 0.0123

To ensure a fair and reproducible comparison, we use a two-stage evaluation protocol that separates split selection from final testing. First, we train each model on all five folds and choose the most representative fold using the class-distribution deviation metric M_j (Equation (6)), without using any performance score. This avoids relying on a single split that may be biased. Next, we fix the selected split and repeat training with different random seeds to measure stability and robustness to random initialization. The full workflow (5-fold training, canonical-fold selection, and multi-seed testing) is shown in Figure 6.

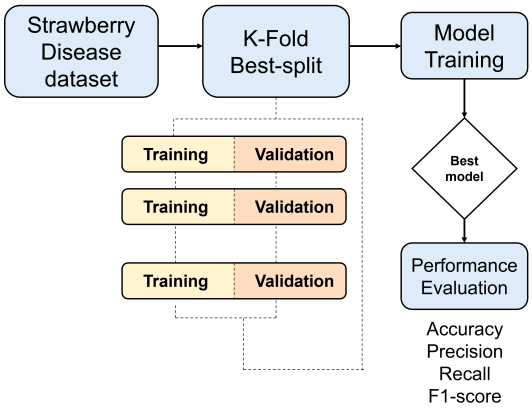


Figure 6. Validation workflow illustrating stratified K-fold partitioning and repeated runs for robustness.

Theoretical rationale and bias considerations. On small and class-imbalanced datasets, random train/test splits may introduce non-negligible label-distribution shift between the overall dataset and the partitions, which increases variance in performance estimation and can destabilize model comparisons. The deviation metric Mj explicitly quantifies this shift via L1 distance on class proportions, and selecting the minimum-deviation fold yields a canonical split that is statistically more representative of the population distribution. This approach reduces sensitivity to idiosyncratic data partitions without requiring the computational cost of full cross-validation averaging. We emphasize that selection is performed using only label proportions (not model performance), thus avoiding performance-driven data leakage. However, because the canonical split is chosen to be distribution-representative, it may be less "adversarial" than an arbitrary split and could yield slightly optimistic estimates compared to worst-case partitions; results should therefore be interpreted as performance on a stable, representative split, with stability further corroborated by repeated-seed experiments. The best-fold strategy used to build StrawberryDualKFold is shown in Algorithm 1. We first compute the overall class-frequency vector of the full dataset, which we use as the target distribution. Next, we generate a standard

5-fold split. For each fold, we measure how much the class proportions in its training and test subsets differ from the target distribution by summing the absolute differences. We then choose the fold with the smallest deviation score and export its images into the usual train/test folder structure. This approach differs from standard repeated cross-validation in two ways. First, repeated cross-validation typically uses the same split with different seeds, ignoring split quality variation. Second, standard K-Fold averages all folds, masking poor split effects. Our best-split selection explicitly optimizes for distributional representativeness before assessing seed stability.

Algorithm 1: Best-Split K-Fold Selection

Input: Dataset $D = \{(x_i, y_i)\}$, $y_i \in \{1, \ldots, C\}$; number of folds $K = 5$
Output: Optimal fold j^* and corresponding $D_{\text{train}}, D_{\text{test}}$
for *each class* $c = 1, \ldots, C$ **do**
  $N_c \leftarrow |\{i : y_i = c\}|$;
  $r^c \leftarrow N_c / N$;
$M_{\text{best}} \leftarrow +\infty$, $j^* \leftarrow -1$;
Split D into K folds: $\{F_1, \ldots, F_K\}$;
for *fold* $j = 1, \ldots, K$ **do**
  Let $(D^j_{\text{train}}, D^j_{\text{test}})$ be the training/test sets;
  for *each subset* $\in \{train, test\}$ **do**
    for *class* $c = 1, \ldots, C$ **do**
      $N^c_{j,\text{subset}} \leftarrow |\{i \in D^j_{\text{subset}} : y_i = c\}|$;
      $r^c_{j,\text{subset}} \leftarrow N^c_{j,\text{subset}} / |D^j_{\text{subset}}|$;
  $M_j \leftarrow \sum_{c=1}^{C} |r^c_{j,\text{train}} - r^c| + \sum_{c=1}^{C} |r^c_{j,\text{test}} - r^c|$;
  if $M_j < M_{best}$ **then**
    $M_{\text{best}} \leftarrow M_j$, $j^* \leftarrow j$;
$D_{\text{train}} \leftarrow D^{j^*}_{\text{train}}$, $D_{\text{test}} \leftarrow D^{j^*}_{\text{test}}$;
return j^*, D_{train}, D_{test}

3.8. Fusion Module and Optimization Strategy

The dual-branch design relies on an adaptive fusion mechanism to integrate complementary local and global representations. Let $F_{\text{local}} \in \mathbb{R}^{B\times H\times W\times C}$ and $F_{\text{global}} \in \mathbb{R}^{B\times H\times W\times C}$ denote the feature maps generated by the DMSCM and Transformer encoder, respectively. Both are spatially aligned (e.g., via bilinear interpolation) and concatenated along the channel axis. A 1×1 convolution followed by a sigmoid activation produces a gating map $g \in [0, 1]^{B\times H\times W\times C}$ that weights each location/channel. FusionGate is applied at two resolutions (32×32 and 16×16 in our implementation). Each resolution uses independently learned gating parameters (1×1 convolutions with 16 and 32 output channels, respectively), though both follow the same mathematical formulation (concatenation, MLP reduction, sigmoid activation). This allows the gate to adapt to different feature scales and channel dimensions at each stage. The output channel count of each 1×1 convolution is set equal to the input feature map channel count C (i.e., 16 in Stage 1 and 32 in Stage 2), ensuring that the fused representation maintains consistent dimensionality with the input branches. The fused representation is:

$$F_{\text{fused}} = g \odot F_{\text{local}} + (1 - g) \odot F_{\text{global}}, \tag{7}$$

where $\odot$ denotes element-wise multiplication. FusionGate is applied hierarchically at two resolutions (e.g., 32×32 and 16×16). To stabilize training, fusion outputs are batch-normalized and passed through residual connections; dropout is applied after the second fusion stage.

An auxiliary supervision term can be applied at the first fusion stage to encourage stable early-layer learning [35]. If used, an auxiliary classifier predicts the disease label from $F_{\text{fused}}^{(1)}$ to produce an auxiliary loss $\mathcal{L}_{\text{aux}}$, and the total objective becomes:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{total}} + \alpha \mathcal{L}_{\text{aux}}, \tag{8}$$

where α controls the auxiliary contribution.

Although both FusionGate instances use the same gating formulation, they do not share parameters—each learns independent 1×1 convolution weights suited to its resolution (16 channels at 32×32, 32 channels at 16×16). Consequently, their functional roles differ: at the earlier fusion stage (higher spatial resolution, e.g., 32×32 for a 128×128 input), the gate operates on fine-grained feature maps where lesion texture and subtle color cues are still preserved; here, the gate mainly blends complementary information so that local morphological details can be reinforced by the Transformer's contextual guidance without losing spatial precision. At the later fusion stage (lower resolution, e.g., 16×16), the features become more semantic and have larger effective receptive fields; in this stage, the gate tends to emphasize discriminative disease patterns while suppressing background clutter and ambiguous context. Therefore, applying FusionGate at multiple depths provides hierarchy-aware fusion: early fusion supports sensitivity to small lesions, while deeper fusion improves robustness through global structure. Figure 7 illustrates the structure of the proposed FusionGate module.

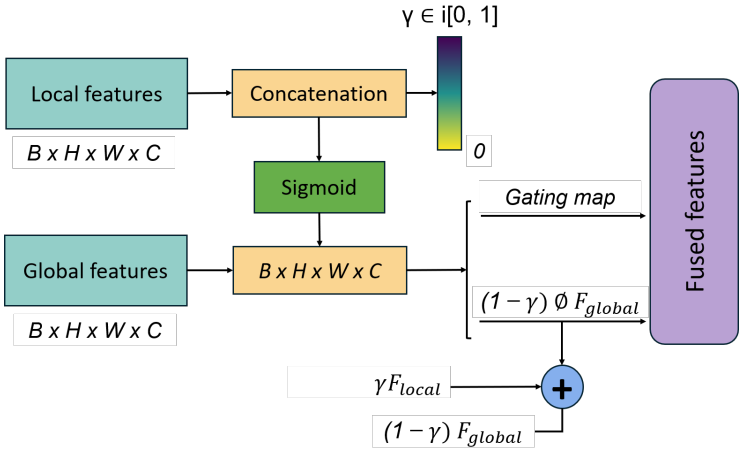


Figure 7. Structure of the proposed FusionGate module. Green boxes represent input feature maps (local and global features), yellow box indicates the concatenation operation, the color gradient bar represents gating coefficient γ ranging from 0 (purple) to 1 (yellow), and the purple box represents the output fused features. The gating map adaptively blends local and global representations through element-wise multiplication.

3.9. Layer Configuration (Compact Design)

StrawberryDualNet is a lightweight convolution–Transformer hybrid comprising a compact stem, two successive dual-branch feature extraction stages, and a minimal classification head. In the stem, the input ($128 \times 128 \times 3$) passes through a 3×3 convolution (stride 2, eight filters), batch normalization, and ReLU, reducing spatial size to 64×64.

From this shared representation, two branches operate in parallel. The local branch applies depthwise-separable convolutions (3×3, 5×5, 7×7) and fuses them using adaptive

coefficients $(\alpha_3, \alpha_5, \alpha_7)$. The global branch processes patch tokens through a lightweight Transformer encoder to capture long-range dependencies. A learned FusionGate merges the two streams to yield a balanced representation. A second extraction stage repeats the design with increased channel capacity. The final feature map is regularized, globally averaged, and passed through a compact dense head ($32 \rightarrow 5$) to output disease probabilities.

3.10. Image Processing and Disease Localization

Image processing provides a post hoc, heuristic localization stage to delineate lesion regions. This approach is class-dependent and approximate, intended for interpretability rather than precise segmentation. Each predicted disease class triggers fixed HSV thresholds and morphological operations, not learned from data. Each input image is resized to 224×224, filtered using a 5×5 Gaussian kernel, and converted to HSV space. Binary masks are formed using predefined hue–saturation–value thresholds corresponding to colors associated with strawberry tissue and fungal symptoms [36]. Morphological closing fills small gaps and opening removes isolated artifacts, yielding a smooth berry region-of-interest (ROI) [37].

Once the ROI is established, the predicted disease class selects a corresponding threshold configuration to produce a lesion mask m_c. The mask is refined using 3×3 morphological opening/closing to consolidate boundaries. Lesion coverage is computed as:

$$\kappa_c = 100 \times \frac{\text{Area}(m_c)}{\text{Area}(\text{berry})}, \tag{9}$$

providing an approximate severity indicator per image. The workflow is summarized in Figure 8.

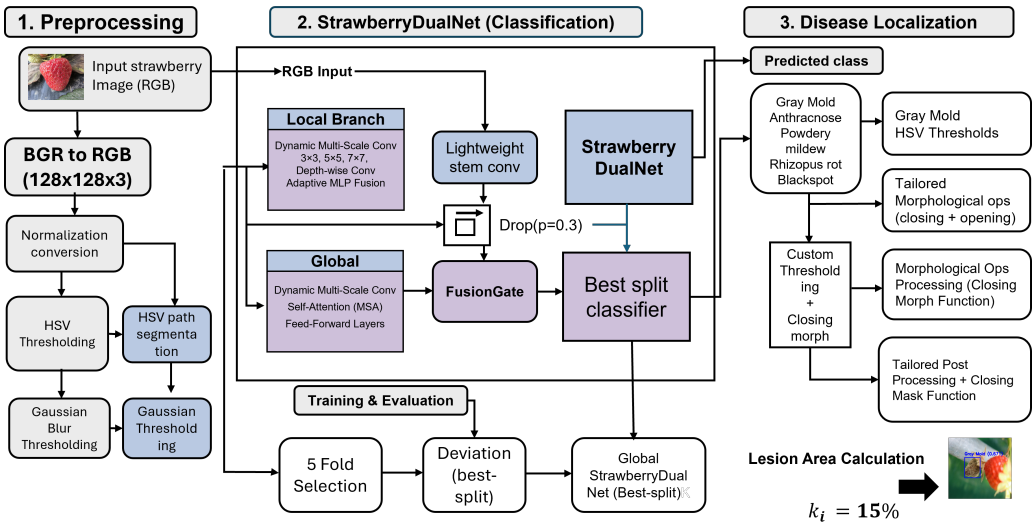


Figure 8. The workflow of StrawberryDualNet classification with HSV- and morphology-based post hoc lesion localization.

Algorithm 2 describes how we classify each strawberry image and estimate lesion coverage. Each RGB image is resized to 224×224 and normalized to $[0, 1]$ by scaling 8-bit pixel values, which improves numerical stability at inference. Classification uses an ensemble of K softmax models $\mathcal{F} = \{m_1, \ldots, m_K\}$; the final probability vector is the mean prediction $\mathbf{p} = \frac{1}{K} \sum_{j=1}^{K} m_j(I)$, which reduces variance compared with a single model. We apply confidence filtering with $\tau = 0.60$: if the maximum confidence $\gamma^\star < \tau$, the image is labeled Healthy and $\kappa = 0\%$. Otherwise, we run a class-specific lesion mapping routine for the predicted disease (e.g., Anthracnose, Gray Mold, Powdery Mildew) then refine

the mask using morphological filtering with a 3×3 opening to remove small noise and a closing step to fill small gaps. Finally, severity is reported as the lesion coverage ratio κ, i.e., the infected area normalized by the fruit area, so it is independent of fruit size in the image.

Algorithm 2: Disease detection and approximate lesion mapping with StrawberryDualNet.

Input : RGB image I; ensemble $\mathcal{F} = \{m_1, \ldots, m_K\}$; threshold τ
Output : Disease label d; lesion coverage κ (%)
$I \leftarrow \text{resize}(I, 224, 224)/255$
$\mathbf{p} \leftarrow \frac{1}{K}\sum_{j=1}^{K} m_j(I)$;
$c^\star \leftarrow \arg\max_c p_c$;
$\gamma^\star \leftarrow \max_c p_c$
if $\gamma^\star < \tau$ **then**
  return *Healthy*, 0
$d \leftarrow \text{classNames}[c^\star]$
$m \leftarrow \text{mask}_d(I)$ `// disease-specific; default = Black Spot` $m \leftarrow \text{close}(\text{open}(m))$
$\kappa \leftarrow 100 \times \dfrac{\text{Area}(m)}{\text{Area}(\text{berry})}$
return d, κ

The lesion mapping stage is implemented as a lightweight post hoc routine based on fixed HSV thresholding and morphological filtering. As a result, it is heuristic rather than learned, and its behavior depends on handcrafted color ranges and structuring-element operations. This design is sensitive to acquisition conditions: illumination variation (over-/under-exposure, shadows), camera white-balance differences, fruit glossiness, and cultivar-dependent skin color can shift HSV distributions and degrade both ROI extraction and lesion masking. The pipeline may also fail when background regions exhibit chromatic properties similar to symptoms, producing false positives. Typical failure cases include specular highlights on glossy berries being misidentified as bright fungal regions (e.g., powdery-mildew-like patterns), low-light shadows obscuring dark lesions (e.g., black spot or anthracnose), and complex scenes with occlusions or multiple fruits that corrupt ROI estimation and bias the coverage computation. In addition, very small lesions may be removed by morphological opening, leading to underestimated severity, while mixed infections or atypical symptom appearance may not match the predefined thresholds. Consequently, the lesion coverage ratio κ should be interpreted as an approximate severity proxy for visual guidance and relative comparison under similar capture conditions, rather than a ground-truth segmentation measure.

4. Results

This section reports the experimental outcomes of the proposed StrawberryDualNet framework, including quantitative comparisons against representative CNN baselines, robustness analyses under multiple random seeds, qualitative visualization of learned feature spaces and confusion patterns, and an ablation study to isolate the contribution of each architectural component. The evaluation focuses on practical strawberry disease monitoring under limited and imbalanced data, emphasizing both performance and stability.

4.1. Experimental Setup

4.1.1. Evaluation Metrics

To assess the performance of StrawberryDualNet and its best-split variant StrawberryDualKFold, we report standard multi-class classification metrics that are widely adopted in

agricultural disease recognition studies. Let TP, FP, TN and FN denote true positives, false positives, true negatives and false negatives, respectively.

Accuracy measures the overall proportion of correctly classified samples:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{10}$$

Precision evaluates the reliability of positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{11}$$

Recall (sensitivity) measures the fraction of true positives that are correctly detected:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{12}$$

$\boldsymbol{F_1}$ score summarizes the balance between precision and recall and is informative under class imbalance:

$$F_1 = 2 \cdot \frac{\text{Precision Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

Beyond scalar metrics, we analyze the separability of learned feature embeddings using t-SNE and UMAP. t-SNE minimizes the Kullback–Leibler divergence between high- and low-dimensional pairwise affinities [38]:

$$C_{\text{t-SNE}} = \sum_{i \neq j} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right), \tag{14}$$

where p_{ij} and q_{ij} denote similarities in the original and embedded spaces. UMAP complements t-SNE by preserving both local and global manifold structure through a cross-entropy objective [39]:

$$L_{\text{UMAP}} = \sum_{i<j} \left[\omega_{ij} \log \left(\frac{\omega_{ij}}{\phi(y_i, y_j)} \right) + (1 - \omega_{ij}) \log \left(\frac{1 - \omega_{ij}}{1 - \phi(y_i, y_j)} \right) \right]. \tag{15}$$

4.1.2. Implementation Details

All experiments were carried out in TensorFlow 2.14 and run on a single NVIDIA T4 GPU (16 GB). Models were trained end-to-end with the Adam optimizer [40], starting from a learning rate of 1×10^{-3}. Input images were uniformly resized to 128×128 pixels and scaled to the $[0, 1]$ range. We used a batch size of 64 for every experiment. Training proceeded for up to 500 epochs with early stopping based on validation accuracy (patience of 10), and the parameters from the epoch achieving the highest validation accuracy were automatically restored.

To reduce randomness from initialization, data shuffling, and mini-batch sampling, we trained each model setting with 30 different random seeds (Within a range of 0 to 30). For StrawberryDualKFold, we used only the canonical best-split fold as the training fold. For StrawberryDualNet, we also report seed-ensemble results by averaging the softmax outputs from the 30 runs. The Dynamic Multi-Scale module uses three depthwise convolution branches (3×3 , 5×5 , 7×7) and an MLP that computes sample-based fusion weights. The Transformer blocks use 4 attention heads. We apply dropout of 0.3 before global pooling. The classifier head includes a 32-unit ReLU layer and a 5-class softmax output layer.

Table 5 provides an overview of the principal hyper-parameters used in the training setup.

Table 5. Hyper-parameter settings of the proposed StrawberryDualNet model.

Parameter	Value
Input image size	128×128
Optimizer	Adam
Initial learning rate	0.001
Batch size	64
Maximum epochs	500
Early stopping patience	10 epochs
Dropout rate	0.3
Number of random seeds	30 (range 0–30)

4.2. Comparative Performance Against Baseline CNNs

We benchmark StrawberryDualNet against seven representative convolutional architectures under an identical training and evaluation protocol. The baseline set includes lightweight mobile-oriented models (MobileNetV2, MobileNetV3, ShuffleNetV2, SqueezeNet), a classical CNN (AlexNet), a residual backbone (ResNet-50), and a high-quality reference network (InceptionV3). To ensure statistical reliability, all architectures were evaluated across 30 independent random seeds, reporting mean performance with 95% confidence intervals. The results are summarized in Table 6.

Table 6. Performance comparison between StrawberryDualNet and baseline CNN architectures. Accuracy is reported as mean ± standard deviation across 30 random seeds with 95% confidence intervals. The columns Accuracy, Precision, Recall, and F1 correspond to the metrics obtained from the single best-performing run among the 30 seeds.

Model	Params	Acc. (Mean ± Std)	95% CI	Accuracy	Prec.	Rec.	F1
StrawberryDualNet	38,595	0.916 ± 0.016	[0.910, 0.919]	0.951	0.953	0.947	0.950
StrawberryDualNetKFold	38,595	0.919 ± 0.018	[0.912, 0.926]	0.951	0.953	0.951	0.952
InceptionV3	2,000,000	0.919 ± 0.017	[0.912, 0.925]	0.949	0.949	0.949	0.949
SqueezeNet	121,701	0.900 ± 0.023	[0.891, 0.909]	0.937	0.944	0.935	0.940
AlexNet	2,997,061	0.806 ± 0.040	[0.791, 0.821]	0.868	0.873	0.855	0.864
MobileNetV2	2,299,141	0.483 ± 0.058	[0.461, 0.505]	0.576	0.778	0.285	0.417
MobileNetV3	957,749	0.476 ± 0.079	[0.446, 0.506]	0.593	0.848	0.308	0.451
ShuffleNetV2	1,274,909	0.849 ± 0.016	[0.843, 0.855]	0.888	0.901	0.889	0.889
ResNet-50	23,653,445	0.884 ± 0.015	[0.879, 0.890]	0.910	0.963	0.812	0.881

Statistical analysis: StrawberryDualNet achieves statistically equivalent accuracy to InceptionV3 (paired t-test: $t = 0.027$, $p = 0.979$, Cohen's $d = 0.005$, n.s.), but with $50\times$ fewer parameters (0.04 M vs. 2.0 M). Significant improvements over other baselines were confirmed: SqueezeNet ($p < 0.01$, $d = 0.58$), AlexNet ($p < 0.001$, $d = 2.45$), and all lightweight/mobile networks ($p < 0.001$, large effect sizes). Notably, StrawberryDualNet outperformed MobileNetV2/V3 and ResNet-50 by substantial margins (>40% absolute accuracy improvement) despite being 50–500$\times$ smaller. The best single-run values demonstrate competitive potential under optimal initialization; these optimistic outliers fall outside the 95% CIs but validate peak performance capability.

To check whether performance is consistent across disease categories, we computed the standard deviation of per-class accuracies (Table 7). StrawberryDualNet has the lowest standard deviation (0.0191), which means its accuracy is stable across all five diseases. By contrast, MobileNetV2 and MobileNetV3 show much larger variation (0.1684 and 0.1501), indicating that they perform well on some diseases but poorly on others (e.g., Powdery Mildew recall of 26.63% for MobileNetV2). These results show that StrawberryDualNet

improves performance in a balanced way across the full task, not only for a small subset of diseases.

Table 7. Per-category accuracy and standard deviation of accuracy across disease categories.

Model	Black Spot	Rhizopus Rot	Anthracnose	Gray Mold	Powdery Mildew	Std. Acc.
StrawberryDualNet	0.9778	0.9884	0.9531	0.9416	0.9740	0.0191
StrawberryDualKFold	0.9552	0.9901	0.8858	0.8578	0.9069	0.0533
InceptionV3	0.9567	0.9877	0.9016	0.8514	0.8972	0.0536
SqueezeNet	0.9640	0.9782	0.8952	0.8375	0.8290	0.0692
ShuffleNetV2	0.9237	0.9564	0.8117	0.7481	0.8363	0.0846
AlexNet	0.8188	0.9164	0.8343	0.7874	0.6728	0.0884
ResNet50	0.9000	0.9901	0.9223	0.8763	0.8600	0.0508
MobileNetV3	0.3989	0.7023	0.5454	0.3399	0.3779	0.1501
MobileNetV2	0.4041	0.6719	0.6291	0.4258	0.2663	0.1684

Seed Sensitivity and Learning Dynamics

Random initialization can affect how the model converges when training on a limited agricultural dataset. Figure 9 compares the training and validation accuracy curves for the best and worst random seeds of StrawberryDualNet and its K-Fold variant. For the weakest seeds, training accuracy quickly reaches 100% while validation accuracy stays around 88–89%, indicating overfitting. For the strongest seeds, the curves remain closely aligned and validation accuracy reaches about 94.5–95.3%.

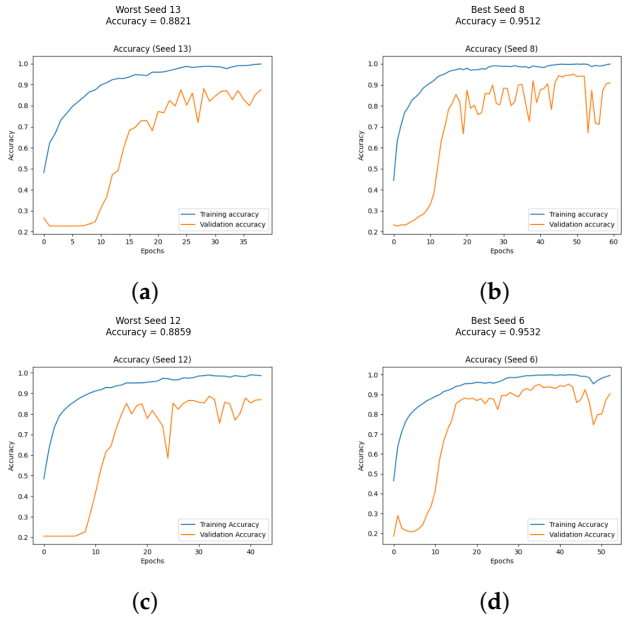


Figure 9. Training and validation accuracy for the worst and best random seeds of StrawberryDualNet (**top row**) and its K-Fold variant (**bottom row**). Y-axis values represent accuracy as proportions (0.0–1.0), corresponding to 0–100%. (**a**) SDN—Worst seed (Acc: 0.8821); (**b**) SDN—Best seed (Acc: 0.9512); (**c**) SDN-KFold—Worst seed (Acc: 0.8859); (**d**) SDN-KFold—Best seed (Acc: 0.9532).

Model performance can also vary across seeds, so we report the full distribution of results rather than only the mean. Figure 10 shows boxplots of Accuracy, Precision, Recall, and F1 Score across 30 random seeds for all evaluated models. {StrawberryDualNet is the most stable model, with a small interquartile range and high median values for all metrics. Its performance variance is low (SD = 1.8%, CV = 1.96%), suggesting that the dual-stream fusion reduces sensitivity to weight initialization. In contrast, some baselines (e.g., MobileNetV2 and ResNet-50) show wider spreads and longer whiskers, meaning

their final performance depends more on the chosen seed, which is less reliable for field deployment. InceptionV3 and SqueezeNet are more stable than these baselines but still do not reach the same overall precision–recall balance.

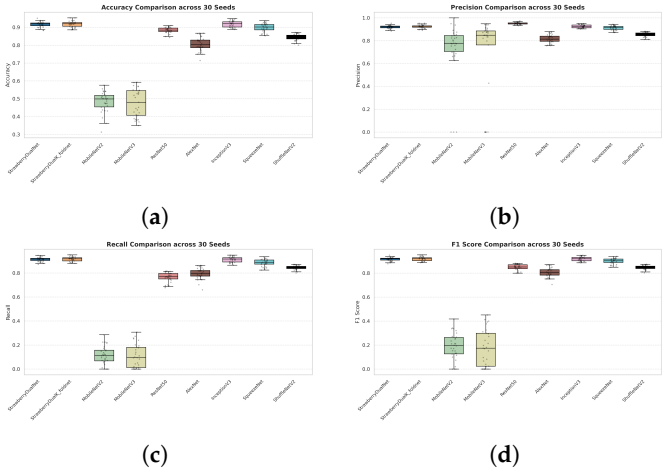


Figure 10. Boxplots showing the distribution of each performance metric (Accuracy, Precision, Recall, F1 Score; all scaled 0–1, i.e., 0–100%) for all evaluated models across 30 random seeds. (**a**) Accuracy; (**b**) Precision; (**c**) Recall; (**d**) F1 Score.

4.3. Computational Efficiency and Feasibility for Edge Deployment

Although classification accuracy remains a primary indicator of diagnostic performance, the practical adoption of deep learning methods in precision agriculture is frequently constrained by hardware limitations. In real-world settings, field deployment commonly relies on edge devices such as smartphones or Raspberry Pi–based scouting platforms, which typically offer limited memory and heterogeneous computational resources. To reflect these operational constraints, we performed a systematic benchmarking study to evaluate the computational efficiency of StrawberryDualNet in comparison with seven widely used deep learning architectures. All models were tested on a common hardware platform (x86_64 CPU @ 2.00 GHz).

It is important to note that these latency and throughput figures are reference metrics on x86_64; achieved performance will differ greatly on ARM-based edge devices (e.g., Raspberry Pi), Jetson-class devices, and mobile SoCs depending on the inference backend, delegates, and memory/IO constraints.

4.3.1. Model Complexity and Storage Efficiency

The primary advantage of the proposed StrawberryDualNet lies in its compact model size. As reported in Table 8, it contains only 0.04 million parameters, corresponding to a storage requirement of just 0.72 MB. This represents a substantial reduction compared to commonly used mobile backbones: StrawberryDualNet is approximately 13× smaller than MobileNetV2 (9.44 MB) and 20× smaller than ShuffleNetV2 (14.99 MB). In rural agricultural environments, where network connectivity is frequently limited or unstable (e.g., 3G/4G), such a small footprint enables faster Over-The-Air (OTA) updates and significantly reduces storage demands on farmers' mobile devices. In addition to parameter count, we report FLOPs to describe compute cost (Table 8), measured with TensorFlow Profiler at 128 × 128 input. StrawberryDualNet needs 71.02 M FLOPs, higher than MobileNetV3 (38.57 M) but far lower than MobileNetV2 (200.23 M), InceptionV3 (535.06 M), and ResNet50 (2531.24 M). The higher FLOPs are expected because our model uses multi-scale parallel convolutions and Transformer attention. Still, the memory footprint stays very small (0.72 MB), which is often the main constraint for edge deployment and model updates.

Table 8. Comprehensive comparison of model complexity, FLOPs, storage size, and CPU inference latency. The proposed model achieves the lowest storage footprint while maintaining competitive inference speeds against modern lightweight baselines.

Model	Params (M)	FLOPs (M)	Size (MB)	CPU Latency (ms)	CPU FPS
StrawberryDualNet (Ours)	0.04	71.02	0.72	168.72 ± 28.95	5.9
MobileNetV3	0.96	38.57	4.12	169.99 ± 6.66	5.9
MobileNetV2	2.30	200.23	9.44	210.50 ± 28.46	4.8
ShuffleNetV2	1.27	58.89	14.99	215.51 ± 9.64	4.6
SqueezeNet	0.12	112.87	1.51	32.53 ± 1.85	30.7
InceptionV3	2.00	535.06	23.07	94.63 ± 15.45	10.6
AlexNet	3.00	317.55	34.36	20.45 ± 1.30	48.9
ResNet50	23.65	2531.24	91.12	340.94 ± 39.58	2.9

4.3.2. Inference Latency Analysis

We evaluated inference latency (forward-pass time) on a reference x86_64 CPU to conduct an edge feasibility assessment for real-time scouting scenarios. StrawberryDualNet achieved an average inference time of 168.72 ms, corresponding to approximately 6 FPS. We note that absolute latency/throughput can vary across target edge hardware (e.g., ARM-based Raspberry Pi, Jetson-class devices, or mobile SoCs) and across deployment stacks and delegates.

Although older and simpler architectures such as AlexNet (20.45 ms) and SqueezeNet (32.53 ms) offer higher raw throughput, their representational capacity is typically insufficient for the complex visual conditions encountered in field environments. More importantly, StrawberryDualNet surpasses several modern lightweight architectures explicitly designed for mobile deployment:

- It is 20% faster than MobileNetV2 (210.50 ms).
- It is 21% faster than ShuffleNetV2 (215.51 ms).
- It attains latency comparable to MobileNetV3 (169.99 ms) while being 5.7× smaller in storage.

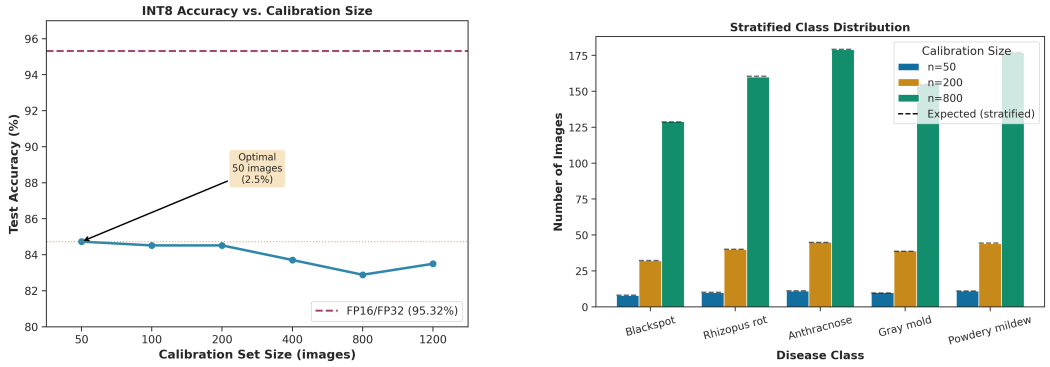
A throughput of ≈6 FPS is insufficient for high-speed video processing, but it is acceptable for "stop-and-go" robotic harvesting and handheld diagnostic use cases, in which response times below 0.2 s are often perceived as practically responsive by users. The benchmarking results further indicate that, for models of this scale, CPU execution can be efficient on the reference platform; however, the benefit of dedicated accelerators and the achievable speedups depend on the target device and software stack. Overall, these findings demonstrate that StrawberryDualNet attains a favorable trade-off between architectural compactness and operational speed, supporting its feasibility as a cost-effective edge solution for smart farming applications, subject to hardware-specific validation. FLOPs do not directly predict latency. Although StrawberryDualNet has about 1.8× more FLOPs than MobileNetV3 (71.02 M vs. 38.57 M), their CPU latency is similar (168.72 ms vs. 169.99 ms, Table 8). This is because latency also depends on memory access, operator implementation, and hardware use. Our depthwise separable operations are often efficient in TensorFlow Lite, so FLOPs and latency should be interpreted together.

4.3.3. Quantized Inference for Edge Deployment

In order to evaluate the feasibility of running the trained networks on-device, we exported the networks to TensorFlow Lite (TFLite). We evaluated networks for both accuracy retention and runtime under bit-width (precision) reduction. In this case, we chose to evaluate two strategies for quantization of the networks: (1) FP16 weight quantization

which means using float16 weights while performing the remaining operations in floating point, and (2) INT8 post-training quantization (PTQ) using a representative calibration set.

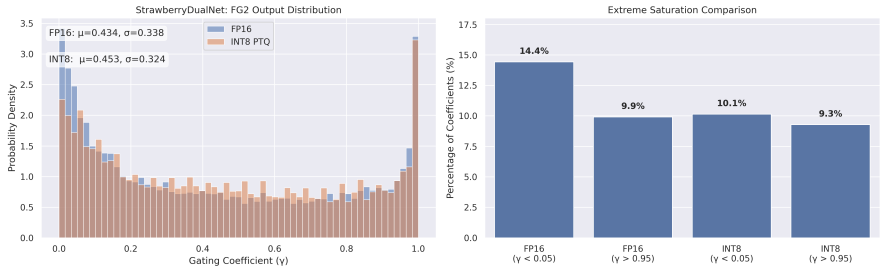
For StrawberryDualNet, the FP16 model matched the baseline Keras performance (Accuracy: 95.32%, F1: 0.9533) and the model size was reduced to 0.127 MB. Calibration set size was systematically optimized via stratified sampling from training data. Testing 50–1200 images revealed peak INT8 accuracy (84.73%) at 50 images (2.5% of training data), with larger sets (400–1200 images) degrading to ∼82.9–83.5%. This counter-intuitive result occurs because larger calibration sets include outlier images that expand the dynamic range (min/max values), reducing precision for normal features critical to dynamic gating. Thus, 50 images provide optimal efficiency-accuracy balance for this architecture (Figure 11).



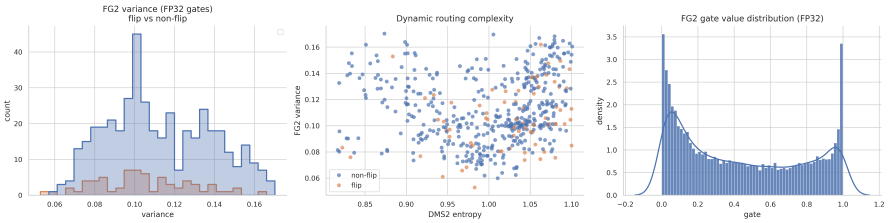
(**a**) (**b**)

Figure 11. Calibration set optimization for INT8 post-training quantization. (**a**) Peak accuracy at 50 images with degradation at larger sets due to outlier-induced dynamic range expansion. INT8 accuracy versus calibration set size, showing peak performance at 50 images (2.5% of training data). The blue solid line represents test accuracy (%) across calibration set sizes, the yellow dotted horizontal line indicates the FP16/FP32 baseline accuracy (95.32%), and the black solid line with arrow highlights the optimal calibration size. (**b**) Stratified sampling maintains class balance across all calibration sizes. Class distribution across calibration sets (n = 50, 200, 800) confirms stratified sampling preserves class proportions; dashed lines indicate expected counts.

However, the size of the model was noticeably worse in the case of INT8-PTQ (Accuracy: 84.52; F1: 0.8393). This degradation corresponds to 68/491 (13.85%) prediction flips relative to FP32, while FP16 TFLite preserves original accuracy, indicating the degradation is quantization-driven rather than architectural. To mechanistically explain the INT8 drop, we quantified how post-training quantization perturbs StrawberryDualNet's intermediate routing signals. Using a TAP-exported model to expose FusionGate outputs (without changing predictions), Figure 12a reports a measurable FG2 distribution shift under INT8 (FP16: $\mu = 0.434$, $\sigma = 0.338$; INT8: $\mu = 0.453$, $\sigma = 0.324$) together with a reduction in extreme gating coefficients ($\gamma < 0.05$: 14.4%→10.1%; $\gamma > 0.95$: 9.9%→9.3%), consistent with quantization-induced compression of gating resolution. Figure 12b further localizes the effect to failure cases: flip samples exhibit statistically higher FG2 routing variability compared to non-flip samples (KS = 0.1537, $p < 0.01$ on per-sample gate variance), showing that INT8 errors concentrate in inputs requiring fine-grained adaptive blending. Collectively, these quantitative shifts in gating distributions and flip-conditioned variability support that the INT8 drop is driven by distortion of sample-dependent gating/branch-selection signals rather than global representational collapse.



(**a**)



(**b**)

Figure 12. Quantization perturbs FusionGate-2 routing. (**a**) FG2 distribution shift and fewer extreme gates under INT8. FG2 under FP16 vs. INT8-PTQ (distribution + extreme γ rates); blue represents FP16 and orange represents INT8-PTQ. (**b**) Errors concentrate in samples with higher routing variability. Flip vs. non-flip: higher FG2 variance for failures (KS = 0.1537, $p < 0.01$); bottom left: orange bars indicate flip samples (prediction errors) and blue bars indicate non-flip samples (correct predictions); bottom right: blue line indicates kernel density estimate of FG2 gate values.

This INT8 degradation should therefore be interpreted as deployment-stack dependent: different calibration strategies, operator coverage, and optimized INT8 delegates (e.g., NNAPI/TensorRT/vendor kernels) can materially change both accuracy and speed. This matches the design of StrawberryDualNet, which has two paths: (1) a fusion path that uses a sigmoid-based FusionGate and branch weighting to control how much each path contributes, and (2) a transformer path that mixes features using attention-like operations. When we apply INT8 post-training quantization (PTQ), the intermediate activations are limited to 8-bit values based on the calibration set, which adds quantization noise. This can also change the scaling of features and shift the FusionGate outputs, so the model becomes less confident between classes and accuracy drops. In contrast, FP16 keeps floating-point computation with a larger value range, so the feature distributions stay closer to training and the learned class boundaries are better preserved.

Runtime results show a clear deployment trade-off. In Table 9, StrawberryDualNet in FP16 runs at 12.0 ms per image on CPU (83.0 FPS), which is suitable for handheld field checks and "stop-and-go" robot inspection. INT8 quantization did not significantly improve inference speed; rather, it increased latency to 69.0 ms compared to FP16's 12.0 ms. This indicates that INT8 speed depends on the deployment runtime and its operator support: when fast INT8 kernels are missing, some layers fall back to slower execution and add overhead. Compared with larger reference models, StrawberryDualNet keeps a strong accuracy–speed balance: it reaches accuracy close to InceptionV3 (94.91%) while being faster on CPU in FP16 (12.0 ms vs. 13.5 ms), and it is far faster than ResNet50 (70.9 ms) while also achieving higher accuracy. FP16 is therefore a practical choice on GPU-capable edge devices (e.g., Jetson-class platforms and modern smartphones) because it keeps accuracy and offers stable speed, while INT8 provides the strongest compression but may reduce accuracy and may not reduce latency without well-optimized INT8 support.

Table 9. Impact of reduced precision on inference speed and diagnostic performance (CPU, batch size = 1). FPS is computed as 1000/latency (ms).

Model	TFLite FP16			TFLite INT8		
	Acc. (%)	Lat. (ms)	FPS	Acc. (%)	Lat. (ms)	FPS
StrawberryDualNet	95.32	12.0	83.0	84.52	69.0	14.5
InceptionV3	94.91	13.5	73.8	94.70	33.1	30.2
SqueezeNet	93.69	3.7	272.7	93.69	3.5	287.5
AlexNet	86.76	6.3	158.5	86.97	8.0	124.5
MobileNetV3	27.70	1.3	777.6	24.24	41.5	24.5
ShuffleNetV2	20.57	10.9	91.6	20.57	6.6	152.4
MobileNetV2	18.33	5.2	193.4	17.92	6.1	164.6
ResNet50	14.66	70.9	14.1	14.46	52.4	19.1

To make the INT8 degradation analysis more specific, we report class-wise accuracy for FP16 and INT8 together with the test sample size n (Table 10). The accuracy drop is not uniform: Anthracnose shows the largest decrease (-24.27 pp, $n = 103$), followed by Gray Mold (-19.59 pp, $n = 97$) and Black Spot (-12.22 pp, $n = 90$). In contrast, Rhizopus Rot shows no change (0.00 pp, $n = 101$), and Powdery Mildew shows a small increase ($+2.00$ pp, $n = 100$). These results suggest that INT8 mainly affects classes with more subtle lesion patterns, while high-contrast symptoms are less affected. The overall decrease (-10.79 pp) is mainly driven by Anthracnose and Gray Mold.

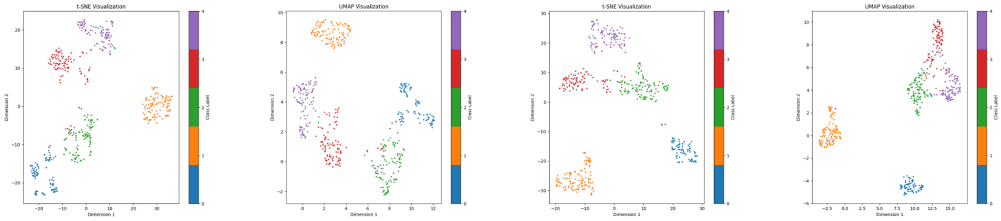
Table 10. Class-wise accuracy under FP16 and INT8 quantization with test sample size n. Drop is reported as percentage-point change (INT8 − FP16).

Disease	n	FP16 Acc. (%)	INT8 Acc. (%)	Drop (pp)
Anthracnose	103	92.23	67.96	−24.27
Gray Mold	97	90.72	71.13	−19.59
Black Spot	90	97.78	85.56	−12.22
Rhizopus Rot	101	100.00	100.00	0.00
Powdery Mildew	100	96.00	98.00	+2.00
Overall	491	95.32	84.52	−10.79

4.4. Qualitative Analysis

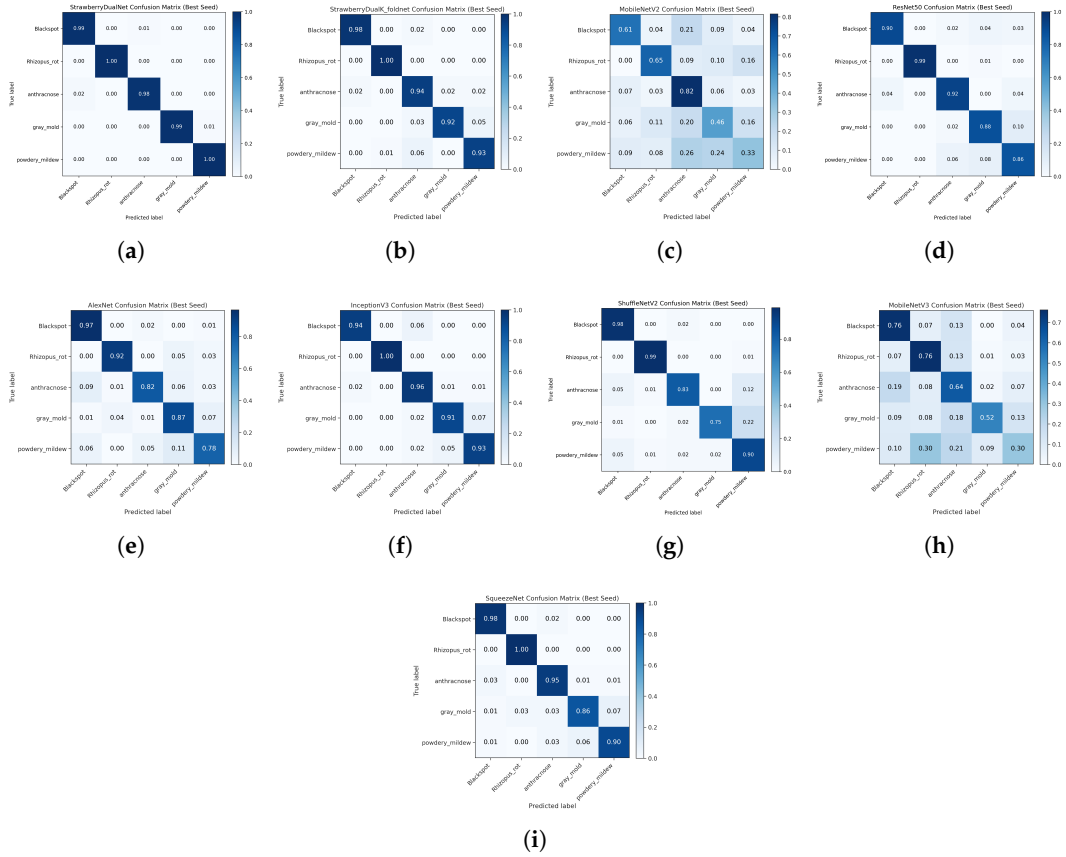
We further evaluate the discriminative capacity of the learned representations using feature-space visualization and confusion-matrix analysis. Figure 13 presents t-SNE and UMAP projections of penultimate-layer embeddings obtained from the best-performing seeds. Both methods reveal well-separated clusters corresponding to the five disease categories, while also reflecting proximity among visually similar classes (e.g., Powdery Mildew versus Gray Mold).

Figure 14 compares normalized confusion matrices across all models. StrawberryDualNet exhibits the strongest diagonal concentration, indicating consistent correct assignments. The K-Fold variant produces slightly fewer off-diagonal errors. Among the baselines, SqueezeNet and InceptionV3 are the closest competitors, whereas lightweight models show substantial cross-class misclassification, reflecting challenges in capturing subtle lesion differences.



(**a**) (**b**) (**c**) (**d**)

Figure 13. Visualization of test-set embeddings for the best-performing seeds: (**a**) t-SNE for StrawberryDualNet, (**b**) UMAP for StrawberryDualNet, (**c**) t-SNE for StrawberryDualNetK-Fold, (**d**) UMAP for StrawberryDualNetK-Fold.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

(**i**)

Figure 14. Normalized confusion matrices for all evaluated models. Compact layout highlights differences in per-class behavior while reducing vertical space. (**a**) SDN; (**b**) SDN-KFold; (**c**) MobileNetV2; (**d**) ResNet-50; (**e**) AlexNet; (**f**) InceptionV3; (**g**) ShuffleNetV2; (**h**) MobileNetV3; (**i**) SqueezeNet.

To make StrawberryDualNet more interpretable, we used Gradient-weighted Class Activation Mapping (Grad-CAM) to highlight which image regions most influenced the model's predictions. In Figure 15, each disease category appears as a pair: the original RGB image on the left and its Grad-CAM heatmap overlay on the right. Following the standard Grad-CAM procedure, we back-propagated the gradient of the target class score to the final convolutional feature maps (the last convolutional layer before the classifier). This gives a coarse, class-discriminative localization map. We then normalized this activation map, upsampled it to the input resolution, and overlaid it on the original image for visual inspection. In these heatmaps, warmer colors (red/yellow) indicate a higher contribution to the predicted class, and cooler colors (blue) indicate a lower contribution. This allows us to check qualitatively whether StrawberryDualNet focuses on symptom-related lesion

patterns rather than on background cues. For consistency, we always computed Grad-CAM from the same final convolutional layer and used the model's own predicted class as the target score for each image. This yields comparable saliency patterns across disease categories and makes it easier to see whether different symptoms activate distinct, localized regions. Such visualization is especially important for strawberry fruit diseases, where lesions can cover only a small portion of the fruit surface and may be partially confused with specular highlights, seeds (achenes), or background vegetation.

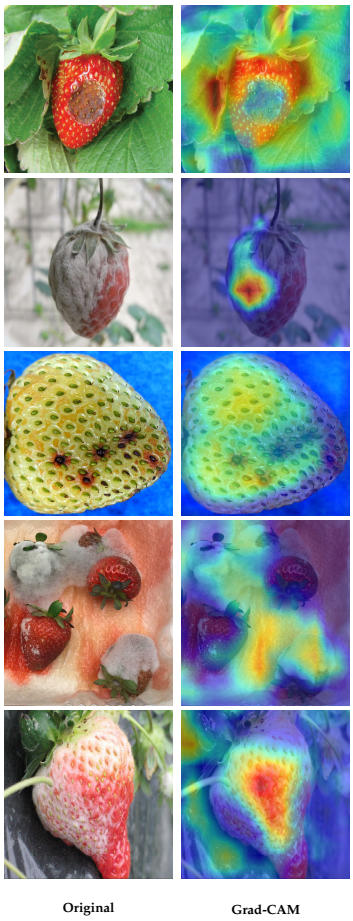


Figure 15. Grad-CAM visualizations of StrawberryDualNet on representative strawberry disease samples. Warmer colors indicate regions contributing more strongly to the predicted class.

The Grad-CAM maps suggest that StrawberryDualNet mainly attends to disease symptoms rather than background. For Powdery Mildew, activations overlap with the whitish mycelial coverage; for Black Spot, the model focuses on dark necrotic lesions around the achenes and largely ignores the blue background; and for Gray Mold, attention concentrates on gray sporulation/mycelium regions. In the multi-fruit Rhizopus/rot scene, attention becomes more spread and partly covers contextual areas (e.g., packaging), which is expected because Grad-CAM provides coarse localization and clutter can introduce background correlations. Overall, the explanations indicate symptom-driven decisions, while also showing that tighter ROI cropping or instance-level analysis may further improve localization robustness.

Figure 16 compares models across the five disease categories. StrawberryDualNet remains focused on lesion regions even with cluttered backgrounds, which helps reduce the gap between predicted and ground-truth IoU. For Anthracnose and Black Spot (rows (a) and (e)), it produces tight boxes around necrotic spots without including healthy tissue, whereas many baselines draw oversized boxes and degrade under blur or leaf

occlusion. For Powdery Mildew (row (c)), several models confuse white fungal regions with background leaves, and for Gray Mold vs. Rhizopus Rot (rows (b) and (d)), which are visually similar, baselines show higher uncertainty. These qualitative results match the quantitative findings and support that the dual-stream fusion improves foreground focus under real field background clutter.

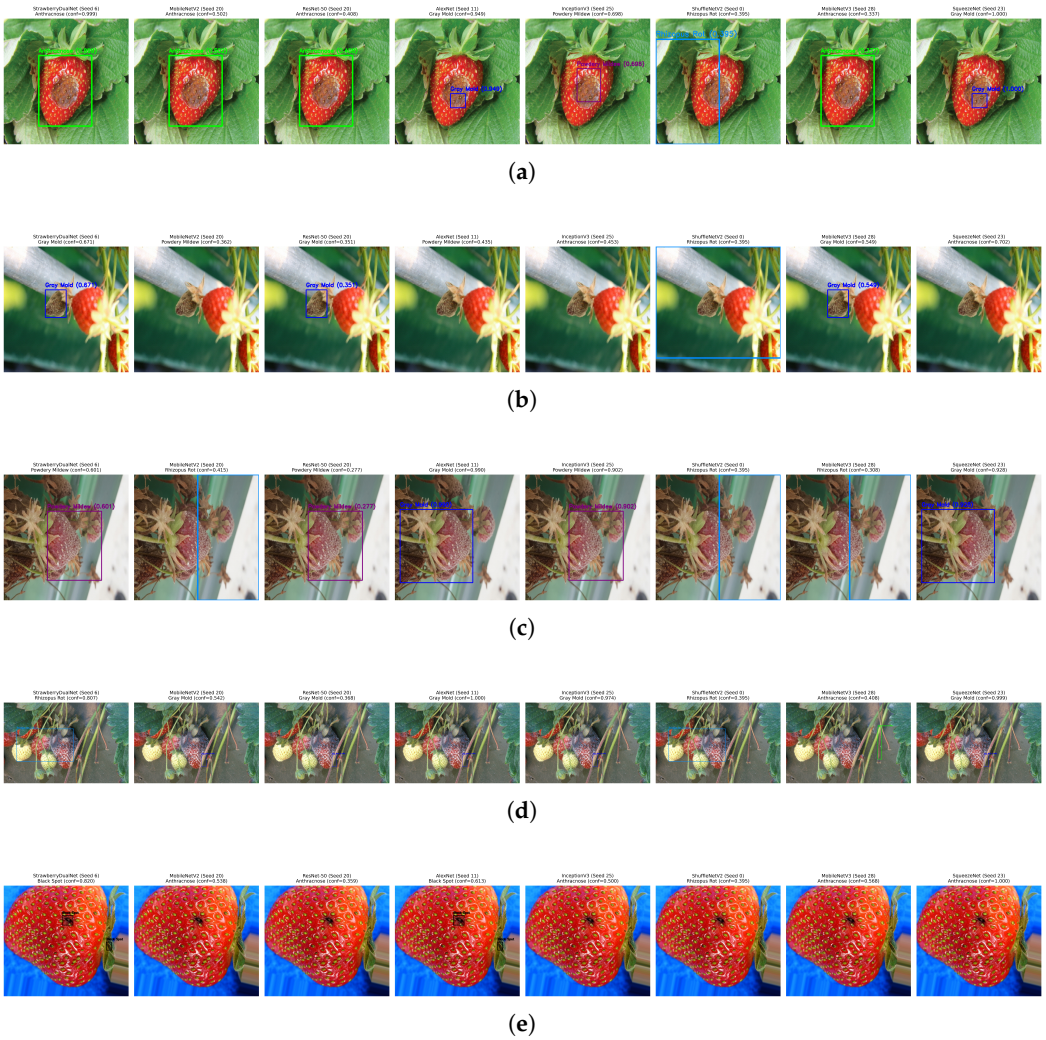


Figure 16. Qualitative comparison of disease localization across all models, showing that StrawberryDualNet consistently provides the most accurate and reliable detections. The bounding boxes are color-coded by disease class, using green for Anthracnose, blue for Gray Mold, purple for Powdery Mildew, cyan for Rhizopus Rot, and black for Black Spot to facilitate visual comparison across models. (**a**) Anthracnose; (**b**) Gray Mold; (**c**) Powdery Mildew; (**d**) Rhizopus Rot; (**e**) Black Spot.

4.5. Ablation Study

An ablation study was conducted to identify which components contribute most to the overall performance of the proposed model [41]. To test which parts of StrawberryDualNet are most important, we removed one component at a time and measured the accuracy drop. We tested four versions: (i) the full model; (ii) without FusionGate (–FG), using simple concatenation instead of adaptive gating; (iii) without Transformer Encoder (–TR), removing the global-context pathway; and (iv) without Dynamic Multi-Scale Convolution (–DMSC), replacing the multi-branch module with a single 3×3 depthwise convolution. Table 11 shows the results for both StrawberryDualNet and StrawberryDualKFold protocols, each with 30 random seeds. The Transformer encoder is the most critical component. Removing it drops accuracy by 0.89 percentage points (StrawberryDualNet) and 1.36 per-

centage points (StrawberryDualKFold). This confirms that global context is essential for distinguishing similar disease symptoms. The Multi-Scale Convolution is also important, with accuracy drops of 1.08 percentage points (StrawberryDualNet) and 0.43 percentage points (StrawberryDualKFold). This shows that adaptive kernel selection helps handle different lesion sizes. The FusionGate has minimal impact in both tests (drops of 0.02 and −0.20 percentage points), meaning simple concatenation works well but gating adds flexibility. Both protocols have in the following in common: the dual-branch design combining local and global features is essential. StrawberryDualKFold gives more stable results, with lower standard deviation in the Multi-Scale test (0.0129 versus 0.0174).

Table 11. Ablation study results for StrawberryDualNet (SDN) and StrawberryDualKFold (SDK) protocols (30 seeds each) with mean metrics: Std = standard deviation of accuracy; 95%CI = 95% confidence interval of accuracy.

	StrawberryDualNet						StrawberryDualKFold					
Variant	**Acc**	**Prec**	**Rec**	**F1**	**Std**	**95% CI**	**Acc**	**Prec**	**Rec**	**F1**	**Std**	**95%CI**
Full model	0.9350	0.9372	0.9335	0.9354	0.0163	0.9290–0.9411	0.9382	0.9411	0.9360	0.9385	0.0172	0.9318–0.9446
No FusionGate (–FG)	0.9348	0.9364	0.9328	0.9346	0.0181	0.9281–0.9416	0.9402	0.9416	0.9379	0.9398	0.0182	0.9334–0.9470
No Transformer (–TR)	0.9262	0.9290	0.9211	0.9250	0.0226	0.9177–0.9346	0.9246	0.9297	0.9194	0.9245	0.0215	0.9166–0.9327
No Multi-Scale (–DMSC)	0.9243	0.9265	0.9217	0.9241	0.0174	0.9178–0.9307	0.9339	0.9361	0.9314	0.9337	0.0129	0.9290–0.9387

5. Discussion

This work was guided by the hypothesis that combining (i) multi-scale convolution features that capture small lesion texture and color changes with (ii) Transformer-based global context should improve strawberry disease recognition when data are limited, and it should also support lesion localization. The results support this idea. The proposed dual-path (CNN–Transformer) design shows higher and more stable performance than single-stream CNN baselines. This suggests that local texture alone is not enough when different fungal diseases look similar and when lighting and background vary. This observation matches recent plant-disease studies reporting that hybrid designs, which mix convolution for local detail and attention for global context, can improve fine-grained disease discrimination [42].

From a strawberry-use point of view, these gains matter because many recent lightweight disease classifiers focus mainly on size and speed, sometimes losing useful visual detail. Cross-domain generalization is important for field use. Our dataset includes both field and greenhouse images, so it already contains changes in lighting, background, and camera distance. The light augmentation (rotation, shift, zoom, and brightness) further reduces sensitivity to capture conditions. However, we did not run a separate external test on unseen farms or strawberry varieties, so performance may drop when fruit appearance or symptom color differs. Compared to StrawberryNet, StrawberryDualNet uses two branches and an adaptive fusion gate to reduce repeated responses while keeping lesion-related cues, which can help reduce confusion between similar fungal symptoms [43]. Compared to BerryNet-Lite, which shows that efficient strawberry recognition is possible on constrained devices, our approach keeps a small parameter footprint while adding an explicit fusion step and a disease-triggered mapping step that improves interpretability for agronomic use [44]. A key point is that high accuracy on a workstation does not always transfer to edge devices. Although StrawberryNet and BerryNet-Lite report strong accuracy, they do not provide a full inference-stage evaluation (e.g., conversion to TFLite/NNAPI/TensorRT and measurements of on-device accuracy, latency/FPS, memory, and energy consumption) to confirm performance under edge constraints. Our experiments show that this step matters: post-training INT8 quantization can reduce accuracy when calibration data are limited and imbalanced, while FP16 tends to preserve accuracy better

and still reduces memory bandwidth and storage. Energy efficiency is important for edge deployment in agriculture, especially when power is limited. On Jetson Nano, StrawberryDualNet with FP16 consumed 2464.73 mJ during inference at 83 FPS, which is about 29.7 mJ per frame (≈0.030 W at 1 FPS). This energy cost is much lower than heavier models and supports our claim that FP16 maintains good accuracy while reducing the deployment load. Overall, the combination of low energy per frame and real-time speed makes FP16 a strong choice for field use. In addition, inference speed (FPS) is important for real workflows such as scouting and sorting; an accurate but slow model may miss short or changing symptoms during continuous capture. For these reasons, accuracy should be reported together with latency and throughput on target hardware, not only with offline metrics.

A second point concerns evaluation reliability. We observed that some architectures are sensitive to weight initialization and small data shifts. This is a common issue in plant-disease modeling, where limited datasets can produce optimistic results from a single split and can lead to unstable model rankings. StrawberryDualKFold addresses a gap in plant disease modeling. Standard practice uses either single splits or simple K-Fold averaging. Our protocol combines best-split selection with multi-seed evaluation. This separates two sources of variance: data partitioning and weight initialization. The result is more stable performance estimates for small, imbalanced agricultural datasets [45].

Beyond classification, lesion localization is important in practice because agronomic actions (e.g., removal, treatment level, sorting) depend on how much of the fruit is affected. Our pipeline links the predicted disease class to a dedicated post-processing routine and produces lesion coverage estimates as a simple severity indicator. This direction is consistent with prior strawberry disease systems that localize symptoms using detection or segmentation, and it supports field interpretability even when pixel-level ground truth masks are not available [46].

Despite these strengths, the study remains limited by (i) dataset scale and diversity (multi-source imagery but limited field variability), (ii) reliance on RGB cues that may miss early or latent infections, (iii) heuristic post-processing whose thresholds may require recalibration across cultivars and lighting conditions, (iv) the fact that the model does not explicitly account for differences in disease symptom expression across strawberry growth stages, including flowering period, young fruit period, and ripening phases (from green to red), which may affect lesion appearance and color-based detection reliability, and (v) and the fact that we did not evaluate on an external cross-variety dataset, so generalization to unseen cultivars and farms still needs validation.

In the broader precision-agriculture context, a natural next step is multimodal sensing. For example, hyperspectral signals can capture biochemical changes before clear RGB symptoms appear, which may help early detection and improve robustness across settings [47]. Future work should focus on the following: (1) Collecting more data under real field conditions across cultivars and growth stage. This should include an external test set collected from different farms, lighting conditions, and strawberry varieties to directly measure cross-domain performance. (2) Testing domain-shift handling (targeted augmentation, adaptation, continual learning). (3) Replacing the heuristic, class-dependent thresholding with weakly supervised or attention-based localization (e.g., using image-level labels or Transformer attention maps) to produce adaptive, learned lesion maps without pixel-level masks or hand-tuned rules. (4) Benchmarking deployment latency and energy on edge hardware to translate experimental gains into practical decision support. And (5) evaluating quantization-aware training (QAT) to mitigate INT8 degradation and further optimize calibration strategies.

6. Conclusions

In this work, we introduced StrawberryDualNet, a compact dual-branch convolutional network that unites dynamic multi-scale convolutions with a lightweight transformer encoder, fused via learnable gates, to perform multi-disease detection on strawberry images. To overcome the challenges posed by our small, imbalanced dataset, we devised a "best-split" K-Fold protocol (StrawberryDualKFold), selecting the most representative partition for training and validation. On the StrawberryDualKFold canonical split, StrawberryDualNet achieved 0.9512 accuracy and 0.9530 precision on disease recognition. Across 30 independent seed runs and up to 500 epochs per run, the model obtained a mean accuracy of 91.9% $\pm$ 1.8% and outperformed deeper architectures such as ResNet-50 and MobileNetV2. In addition to disease-type prediction, we integrated disease-specific image analysis steps, including HSV-based color characterization and morphological filtering, to delineate symptomatic regions for Anthracnose, Gray Mold, Powdery Mildew, Rhizopus Rot, and Black Spot. This combined pipeline reduces manual inspection effort and supports rapid field screening under resource-constrained conditions. The latency results come from an edge-oriented feasibility test on a reference x86_64 CPU. Real-world runtime and INT8 quantization performance can change across different hardware platforms and deployment stacks. Moreover, repeating training across 30 random seeds and applying statistical comparisons increases confidence that the reported performance is stable and not driven by a single favorable initialization.

As we look to the future, we intend to expand our collection with additional images. This will include photos of various types of strawberries, photos showing the fruit in different stages of growth, and images from various weather and agricultural conditions. A broader selection of images will enhance our model's ability to perform well in unfamiliar situations. We might also experiment with advanced sensing technologies, such as cameras that perceive a wider spectrum of colors than the human eye or those that assess heat (like hyperspectral or thermal imaging). Investigating methods to train the model with less hand-labeled data, using semi-supervised or self-supervised learning, could aid in identifying infections that are either very small or just emerging. Our research demonstrates that by carefully developing a lightweight network and strategically partitioning the data, we can achieve results that are comparable to, or even surpass, those of larger models. This indicates that StrawberryDualNet is progressing towards the development of effective systems for intelligent strawberry farming that are efficient and broadly applicable.

Author Contributions: Conceptualization, N.H. and M.Z.; methodology, N.H. and M.S.; software, N.H.; validation, M.E.A., K.E.A. and Y.E.K.; formal analysis, M.S. and H.R.; investigation, N.H. and K.E.A.; resources, L.M.; data curation, N.H. and M.E.A.; writing—original draft preparation, N.H.; writing—review and editing, M.Z., M.S. and L.M.; visualization, N.H. and K.E.A.; supervision, M.Z. and L.M.; project administration, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.kaggle.com/datasets/usmanafzaal/strawberry-disease-detection-dataset (accessed on 15 January 2026).

Acknowledgments: The authors would like to thank all researchers whose work is cited in this manuscript, as well as the anonymous reviewers for their constructive comments and suggestions,

which helped improve the clarity and quality of this work. We also gratefully acknowledge the technical and administrative support that facilitated the completion of this study.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
TE	Transformer Encoder
DMSC	Dynamic Multi-Scale Convolution
FusionGate	Adaptive feature fusion gate
SDN	StrawberryDualNet
SDN-KFold	StrawberryDualNetKFold
K-Fold	K-Fold cross-validation
GAP	Global Average Pooling
MLP	Multi-Layer Perceptron
BN	Batch Normalization
LN	Layer Normalization
ReLU	Rectified Linear Unit
MSA	Multi-Head Self-Attention
FFN	Feed-Forward Network
RGB	Red–Green–Blue
HSV	Hue–Saturation–Value
ROI	Region of Interest
GPU	Graphics Processing Unit
t-SNE	t-distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection

References

1. Du, Y.-c.; Yuan, C.-s.; Song, Y.-q.; Yang, Y.; Zheng, Q.-s.; Hou, Q.; Wang, D.; Wang, L. Enhancing soil health and strawberry disease resistance: The impact of calcium cyanamide treatment on soil microbiota and physicochemical properties. *Front. Microbiol.* **2024**, *15*, 1366814. [CrossRef]
2. Chang, Y.K.; Mahmud, M.S.; Shin, J.; Nguyen-Quang, T.; Price, G.W.; Prithiviraj, B. Comparison of Image Texture Based Supervised Learning Classifiers for Strawberry Powdery Mildew Detection. *AgriEngineering* **2019**, *1*, 434–452. [CrossRef]
3. Yang, Y.; Mali, P.; Arthur, L.; Molaei, F.; Atsyo, S.; Geng, J.; He, L.; Ghatrehsamani, S. Advanced technologies for precision tree fruit disease management: A review. *Comput. Electron. Agric.* **2025**, *229*, 109704. [CrossRef]
4. Toda, Y.; Okura, F. How Convolutional Neural Networks Diagnose Plant Disease. *Plant Phenomics* **2019**, *2019*, 9237136. [CrossRef]
5. Yang, T.; Wang, Y.; Lian, J. Plant Diseased Lesion Image Segmentation and Recognition Based on Improved Multi-Scale Attention Net. *Appl. Sci.* **2024**, *14*, 1716. [CrossRef]
6. Liu, C.; Cao, Y.; Wu, E.; Yang, R.; Xu, H.; Qiao, Y. A Discriminative Model for Early Detection of Anthracnose in Strawberry Plants Based on Hyperspectral Imaging Technology. *Remote Sens.* **2023**, *15*, 4640. [CrossRef]
7. Zhang, M.; Liu, C.; Li, Z.; Yin, B. From Convolutional Networks to Vision Transformers: Evolution of Deep Learning in Agricultural Pest and Disease Identification. *Agronomy* **2025**, *15*, 1079. [CrossRef]
8. Lv, Z.; Yang, S.; Ma, S.; Wang, Q.; Sun, J.; Du, L.; Han, J.; Guo, Y.; Zhang, H. Efficient Deployment of Peanut Leaf Disease Detection Models on Edge AI Devices. *Agriculture* **2025**, *15*, 332. [CrossRef]
9. Yue, X.; Qi, K.; Na, X.; Zhang, Y.; Liu, Y.; Liu, C. Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage. *Agriculture* **2023**, *13*, 1643. [CrossRef]
10. Teodorescu, V.; Obreja Brașoveanu, L. Assessing the Validity of k-Fold Cross-Validation for Model Selection: Evidence from Bankruptcy Prediction Using Random Forest and XGBoost. *Computation* **2025**, *13*, 127. [CrossRef]
11. Zhang, M.; Lin, Z.; Tang, S.; Lin, C.; Zhang, L.; Dong, W.; Zhong, N. Dual-Attention-Enhanced MobileViT Network: A Lightweight Model for Rice Disease Identification in Field-Captured Images. *Agriculture* **2025**, *15*, 571. [CrossRef]
12. Truong-Dang, V.L.; Thai, H.T.; Le, K.H. TinyResViT: A lightweight hybrid deep learning model for on-device corn leaf disease detection. *Internet Things* **2025**, *30*, 101495. [CrossRef]

13. Upadhyay, A.; Chandel, N.S.; Singh, K.P.; Chakraborty, S.K.; Nandede, B.M.; Kumar, M.; Subeesh, A.; Upendar, K.; Salem, A.; Elbeltagi, A. Deep learning and computer vision in plant disease detection: A comprehensive review of techniques, models, and trends in precision agriculture. *Artif. Intell. Rev.* **2025**, *58*, 92. [CrossRef]
14. Xiao, J.R.; Chung, P.C.; Wu, H.Y.; Phan, Q.H.; Yeh, J.L.A.; Hou, M.T.K. Detection of Strawberry Diseases Using a Convolutional Neural Network. *Plants* **2021**, *10*, 31. [CrossRef]
15. Wu, G.; Fang, Y.; Jiang, Q.; Cui, M.; Li, N.; Ou, Y.; Diao, Z.; Zhang, B. Early identification of strawberry leaves disease utilizing hyperspectral imaging combing with spectral features, multiple vegetation indices and textural features. *Comput. Electron. Agric.* **2023**, *204*, 107553. [CrossRef]
16. Li, Y.; Wang, J.; Wu, H.; Yu, Y.; Sun, H.; Zhang, H. Detection of powdery mildew on strawberry leaves based on DAC-YOLOv4 model. *Comput. Electron. Agric.* **2022**, *202*, 107418. [CrossRef]
17. Chen, S.; Liao, Y.; Lin, F.; Huang, B. An Improved Lightweight YOLOv5 Algorithm for Detecting Strawberry Diseases. *IEEE Access* **2023**, *11*, 54080–54092. [CrossRef]
18. Mihajlovic, M.; Marjanovic, M. Enhancing Instance Segmentation in High-Resolution Images Using Slicing-Aided Hyper Inference and Spatial Mask Merging Optimized via R-Tree Indexing. *Mathematics* **2025**, *13*, 3079. [CrossRef]
19. Hu, X.; Wang, R.; Du, J.; Hu, Y.; Jiao, L.; Xu, T. Class-attention-based lesion proposal convolutional neural network for strawberry diseases identification. *Front. Plant Sci.* **2023**, *14*, 1091600. [CrossRef] [PubMed]
20. Yang, B.; Wang, Z.; Guo, J.; Guo, L.; Liang, Q.; Zeng, Q.; Zhao, R.; Wang, J.; Li, C. Identifying plant disease and severity from leaves: A deep multitask learning framework using triple-branch Swin Transformer and deep supervision. *Comput. Electron. Agric.* **2023**, *209*, 107809. [CrossRef]
21. Li, G.; Jiao, L.; Chen, P.; Liu, K.; Wang, R.; Dong, S.; Kang, C. Spatial convolutional self-attention-based transformer module for strawberry disease identification under complex background. *Comput. Electron. Agric.* **2023**, *212*, 108121. [CrossRef]
22. Jia, S.; Wang, G.; Li, H.; Liu, Y.; Shi, L.; Yang, S. ConvTransNet-S: A CNN-Transformer Hybrid Disease Recognition Model for Complex Field Environments. *Plants* **2025**, *14*, 2252. [CrossRef]
23. Li, X.; Li, S. Transformer Help CNN See Better: A Lightweight Hybrid Apple Disease Identification Model Based on Transformers. *Agriculture* **2022**, *12*, 884. [CrossRef]
24. Gookyi, D.A.N.; Wulnye, F.A.; Wilson, M.; Danquah, P.; Danso, S.A.; Gariba, A.A. Enabling Intelligence on the Edge: Leveraging Edge Impulse to Deploy Multiple Deep Learning Models on Edge Devices for Tomato Leaf Disease Detection. *AgriEngineering* **2024**, *6*, 3563–3585. [CrossRef]
25. Zhang, P.; Li, R.; Liu, Y.; Sun, G.; Wen, C. I-GhostNetV3: A Lightweight Deep Learning Framework for Vision-Sensor-Based Rice Leaf Disease Detection in Smart Agriculture. *Sensors* **2026**, *26*, 1025. [CrossRef]
26. Ochoa-Ornelas, R.; Gudiño-Ochoa, A.; Rodríguez González, A.Y.; Trujillo, L.; Fajardo-Delgado, D.; Puga-Nathal, K.L. Lightweight and Accurate Deep Learning for Strawberry Leaf Disease Recognition: An Interpretable Approach. *AgriEngineering* **2025**, *7*, 355. [CrossRef]
27. Afzaal, U.; Bhattarai, B.; Pandeya, Y.R.; Lee, J. An Instance Segmentation Model for Strawberry Diseases Based on Mask R-CNN. *Sensors* **2021**, *21*, 6565. [CrossRef] [PubMed]
28. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
30. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective Kernel Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–21 June 2019; pp. 510–519. [CrossRef]
31. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
32. Zhang, H.W.; Wang, R.F.; Wang, Z.; Su, W.H. DLCPD-25: A Large-Scale and Diverse Dataset for Crop Disease and Pest Recognition. *Sensors* **2025**, *25*, 7098. [CrossRef] [PubMed]
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [CrossRef]
34. Mittal, P.; Tanwar, V.; Sharma, B.; Yadav, D.P. Unleashing the Potential of Residual and Dual-Stream Transformers for the Remote Sensing Image Analysis. *J. Imaging* **2025**, *11*, 156. [CrossRef]
35. Yu, Z.; Zhao, L.; Liao, T.; Zhang, X.; Chen, G.; Xiao, G. A novel non-pretrained deep supervision network for polyp segmentation. *Pattern Recognit.* **2024**, *154*, 110554. [CrossRef]
36. Xu, Y.X.; Yu, X.H.; Yi, Q.; Zhang, Q.Y.; Su, W.H. Dual-Phase Severity Grading of Strawberry Angular Leaf Spot Based on Improved YOLOv11 and OpenCV. *Plants* **2025**, *14*, 1656. [CrossRef]

37. Mu, S.; Liu, J.; Zhang, P.; Yuan, J.; Liu, X. YS3AM: Adaptive 3D Reconstruction and Harvesting Target Detection for Clustered Green Asparagus. *Agriculture* **2025**, *15*, 407. [CrossRef]
38. Zabel, S.; Hennig, P.; Nieselt, K. Visualizing stability: A sensitivity analysis framework for t-SNE embeddings. *Front. Bioinform.* **2026**, *5*, 1719516. [CrossRef]
39. Yi, W.; Bu, S.; Lee, H.H.; Chan, C.H. Comparative Analysis of Manifold Learning-Based Dimension Reduction Methods: A Mathematical Perspective. *Mathematics* **2024**, *12*, 2388. [CrossRef]
40. Bhakta, S.; Nandi, U.; Changdar, C.; Paul, B.; Si, T.; Pal, R.K. aMacP: An adaptive optimization algorithm for Deep Neural Network. *Neurocomputing* **2025**, *620*, 129242. [CrossRef]
41. Tang, H.; Liu, D.; Shen, C. Data-efficient multi-scale fusion vision transformer. *Pattern Recognit.* **2025**, *161*, 111305. [CrossRef]
42. Hu, B.; Jiang, W.; Zeng, J.; Cheng, C.; He, L. FOTCA: Hybrid transformer-CNN architecture using AFNO for accurate plant leaf disease image recognition. *Front. Plant Sci.* **2023**, *14*, 1231903. [CrossRef]
43. Li, X.; Jiao, L.; Liu, K.; Liu, Q.; Wang, Z. StrawberryNet: Fast and Precise Recognition of Strawberry Disease Based on Channel and Spatial Information Reconstruction. *Agriculture* **2025**, *15*, 779. [CrossRef]
44. Wang, J.; Li, Z.; Gao, G.; Wang, Y.; Zhao, C.; Bai, H.; Lv, Y.; Zhang, X.; Li, Q. BerryNet-Lite: A Lightweight Convolutional Neural Network for Strawberry Disease Identification. *Agriculture* **2024**, *14*, 665. [CrossRef]
45. Pacal, I.; Kunduracioglu, I.; Alma, M.H.; Deveci, M.; Kadry, S.; Nedoma, J.; Slany, V.; Martinek, R. A systematic review of deep learning techniques for plant diseases. *Artif. Intell. Rev.* **2024**, *57*, 304. [CrossRef]
46. Chen, M.; Zou, W.; Niu, X.; Fan, P.; Liu, H.; Li, C.; Zhai, C. Improved YOLOv8-Based Segmentation Method for Strawberry Leaf and Powdery Mildew Lesions in Natural Backgrounds. *Agronomy* **2025**, *15*, 525. [CrossRef]
47. Barbedo, J.G.A. A review on the combination of deep learning techniques with proximal hyperspectral images in agriculture. *Comput. Electron. Agric.* **2023**, *210*, 107920. [CrossRef]

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.