Journal of Agricultural Engineering

https://www.agroengineering.org/

Efficient and explainable attention-based multitask learning for fruit type, ripeness, and disease classification

Rida El Chall,[1,2] Sarah Alayan,[1] Abed Ellatif Samhat[1]

[1]Scientific Research Center in Engineering (CRSI), Faculty of Engineering, Lebanese University (LU), Beirut
[2]School of Engineering, Computer and Communications Department, Lebanese International University (LIU), Beirut, Lebanon

Corresponding author: Rida El Chall, Scientific Research Center in Engineering (CRSI), Faculty of Engineering, Lebanese University (LU), Beirut, Lebanon. E-mail: rida.chall@ul.edu.lb

Please cite this article as doi: 10.4081/jae.2026.2108

Submitted: 19 January 2026
Accepted: 10 September 2026

Efficient and explainable attention-based multitask learning for fruit type, ripeness, and disease classification

Rida El Chall,[1,2] Sarah Alayan,[1] Abed Ellatif Samhat[1]

[1]Scientific Research Center in Engineering (CRSI), Faculty of Engineering, Lebanese University (LU), Beirut
[2]School of Engineering, Computer and Communications Department, Lebanese International University (LIU), Beirut, Lebanon

Corresponding author: Rida El Chall, Scientific Research Center in Engineering (CRSI), Faculty of Engineering, Lebanese International University (LIU), Beirut, Lebanon. E-mail: rida.chall@ul.edu.lb

Authors' Contributions: Rida El Chall, writing – review & editing, conceptualization, methodology, software, supervision. **Sarah Alayan**, writing – original draft, conceptualization, methodology, software. **Abed Ellatif Samhat**, writing – review & editing, supervision. All authors read and approved the final version of the manuscript and agreed to be accountable for all aspects of the work.

Competing interests: The authors declare that they have no competing interests.

Data Availability Statement: All the data and materials used are included and referenced in this paper. The source code and implementation details are publicly available at: https://github.com/ml-research-ul/AgriMTL

Abstract

Precision agriculture increasingly relies on automated crop monitoring to support disease prevention and harvest decisions. However, most existing systems treat fruit type, ripeness, and disease classification as separate tasks, leading to high computational costs in real-world deployments. This paper proposes an efficient and explainable multitask learning (MTL) model for fruit monitoring that performs all three tasks simultaneously. The MTL model uses a shared lightweight MobileNetV2 backbone with a convolutional block attention module (CBAM) to extract both shared and discriminative features. Dedicated classification heads then utilize these features to generate predictions for each task. To address the scarcity of fully annotated agricultural datasets, we combine multiple public datasets and apply a partial label-masking strategy to handle missing annotations. In addition, the proposed model is evaluated using several backbone architectures, including ResNet-50, EfficientNetV2-M, and ViT-B/16, under both single optimizer and separate optimizers. Experimental results demonstrate that the proposed MTL model achieves 99.39% (fruit type), 96.35% (ripeness), and 99.7% (disease) accuracy, while using only 5.5M parameters, outperforming single-task solutions. Furthermore, gradient-based explainability analysis provides transparency into the model's decision-making process.

Key words: attention mechanism (CBAM); disease detection; explainable AI; fruit monitoring; MobileNetV2; multitask learning; ripeness detection.

Introduction

The rapid growth of the global population increases pressure on the agricultural sector to improve productivity, maintain crop quality, and reduce losses. Traditionally, tasks such as fruit type classification, ripeness estimation, and disease detection are performed manually. These methods are effective for small farms, but they are labor-intensive, time-consuming, and prone to errors in large-scale deployments.

Recent advances in deep learning have enabled automated solutions in precision agriculture (Attri *et al*., 2023; Padhiary *et al*., 2024). Convolutional neural networks (CNNs) (Hamed *et al*., 2023; Krishna *et al*., 2025), recurrent neural networks (RNNs), attention-based models (Wang *et al*., 2024), and vision transformers (ViTs) (Bai *et al*., 2024) have demonstrated strong performance in disease prediction, ripeness estimation, and fruit classification tasks. Despite these advances, most prior research focuses on task-specific learning, where a separate model is trained for each task. This leads to increased computational cost, redundancy in feature extraction, and inefficiencies in real-world deployment, especially on small edge devices.

Multitask learning (MTL) provides a promising alternative by enabling a single model to learn multiple related tasks simultaneously through shared representations. MTL has been widely adopted in computer vision, robotics, and medical analysis due to its ability to reduce overfitting, improve generalization, and enhance computational efficiency (Yu *et al*., 2024). However, its application in agriculture remains limited. Existing studies typically address only two tasks, such as fruit type and freshness (Zhang *et al*., 2024). Nair *et al*. (2024) proposed a unified CNN-based system for fruit quality assessment and plant disease detection. A key challenge in developing MTL models for agriculture is the lack of fully annotated datasets that simultaneously include fruit type, ripeness, and disease labels.

Therefore, this work aims to develop a lightweight MTL system that jointly performs fruit classification, ripeness estimation, and disease detection under partially labeled data conditions for real-world agricultural applications. The main contributions of this work are as follows:

- A dataset strategy that combines multiple publicly available datasets for joint MTL training while handling missing labels through a loss-masking mechanism.
- A unified MTL model using a single shared backbone (MobileNetV2) with a convolutional block attention module (CBAM) and task-specific heads.
- A systematic evaluation of different backbone architectures (ResNet-50, EfficientNet-V2-M, and ViT-B/16), attention mechanisms, and optimizer strategies (shared optimizer vs. dedicated optimizers per task).

In addition, gradient-weighted class activation mapping (Grad-CAM) (Selvaraju *et al*., 2017) is utilized to enhance model explainability by visually highlighting image regions that contribute to the predictions.

State of the art

This section reviews relevant studies on disease detection, ripeness assessment, fruit type classification, and multitask learning in agriculture.

Plant disease detection

Plant disease detection has been improved through deep learning models, attention mechanisms and transformer-based architectures. EfficientNet and its variants achieve high accuracy on benchmark datasets. For instance, EfficientNet-B4 and B5 (Atila *et al*., 2021) reached 99.91-99.97% accuracy on the PlantVillage dataset (Ali, 2019). CA-ENet (coordinate attention + EfficientNet) achieved 98.92% accuracy on the Apple Leaf Disease Identification Dataset (ALDID) (Wang *et al*., 2021), constructed by augmenting apple images from PlantVillage. However, these results are obtained mainly from controlled environments, limiting real-world applicability. EfficientNetV2S (Tan and Le, 2021) fine-tuned on a NewPlant dataset (Bhattarai, 2018) achieved 95.01% accuracy across 38 disease classes under real field constraints such as noise and illumination variations (Hamed *et al*., 2023). EfficientNet-B3 (Krishna *et al*., 2025) achieved 80.19% accuracy and F1-scores above 90% on PlantDoc (Uddin, 2020) and web-sourced images. The performance gap between controlled and field datasets indicates sensitivity to real-world conditions. Transformer-based models have also shown strong performance, but with greater computational cost. The Swin Transformer-based TSTC model achieved 99.00% accuracy for disease classification and 88.73% for severity prediction on the AI Challenger 2018 dataset (Yang *et al.*, 2023). DFN-PSAN achieved 95.27% accuracy with reduced parameters using pyramidal attention mechanisms across multiple agricultural datasets (Dai *et al.*, 2024). DINOV2-FCS combines local and global features, achieving 99.67% accuracy on a multi-fruit leaf disease dataset (Bai *et al.*, 2024).

Fruit type and ripeness classification

Fruit-type classification and ripeness estimation have been addressed through lightweight CNN architectures such as MobileNet and YOLO variants. MobileNetV2-based systems (Tapia-Mendez *et al.*, 2023) achieved 97.86% accuracy on the Fruit Classification dataset (sshikamaru, 2018) and 100% accuracy for ripeness detection across multiple categories.

A YOLOv8-based model reached 98% accuracy under varying lighting and background conditions, outperforming earlier YOLO versions (Zhang, 2023). Similarly, a hybrid YOLOv9 (Wang *et al.*, 2024) – Swin Transformer model for strawberry ripeness achieved 87.3% mAP at IoU=0.5 by combining local and global feature learning (Mi and Yan, 2024). Despite strong individual performance, these approaches are built as single-task detection or classification pipelines.

Multitask learning

Multitask learning reduces computational cost by using a single model to learn shared representations across related tasks. However, existing MTL studies in agriculture remain limited in scope, typically addressing only two tasks at a time. A CNN-based MTL model (Zhang *et al.*, 2024) achieved 93.24% accuracy for freshness and 88.66% for fruit type classification on a fresh and rotten fruit dataset (Sultana *et al.*, 2022), with no disease detection component. Transformer-based PDLC-ViT achieved near-perfect performance (99.97% accuracy and 99.18% mAP) for joint disease detection and localization (Hemalatha and Jayachandran, 2024), but at high computational cost. More recently, the Plant Disease Multitask Joint Detection Model (PMJDM) (Fu *et al.*, 2025) achieved 71.84% precision and 61.83% mAP50 using ConvNeXt-based shared features, ignoring ripeness and multi-class fruit classification entirely.

Despite significant progress, existing studies address at most two tasks and lack a unified framework that simultaneously integrates fruit classification, ripeness assessment, and disease detection for real-time multi-task deployment.

Table 1 summarizes deep learning models for different agricultural tasks, comparing model architectures, datasets, and performance metrics.

Materials and Methods

This section describes the proposed MTL model that jointly performs three related agricultural tasks: fruit classification, ripeness assessment, and disease detection, as illustrated in Figure 1. First, data from multiple public datasets (PlantDoc, NewPlant Diseases, Fruit Ripeness Level, and Fruit Image datasets) are combined into two composite datasets: a Fruit Disease Dataset and a Fruit Ripeness Dataset. After preprocessing, a shared backbone (MobileNetV2, ResNet-50, EfficientNetV2-M, or ViT-B/16) is used to extract common visual features from images. Subsequently, none, one, or three CBAM attention modules are applied to refine feature maps at the channel and spatial levels. These attention modules help the model capture informative regions and suppress irrelevant regions, allowing it to learn both general and task-specific visual features, such as color, texture, and shape. These shared features are then passed to three task-specific heads.

Dataset construction

A major challenge in MTL for agricultural applications is the absence of datasets that simultaneously provide annotations for fruit type, ripeness, and disease. Existing datasets provide either fruit-and-disease or fruit-and-ripeness labels. To address this limitation, we construct two composite datasets by combining multiple publicly available datasets, each providing partial annotations. Both fruit-disease and fruit-ripeness datasets include four main types of fruit: apples, grapes, oranges, and strawberries. The disease dataset includes both healthy and various common diseases, such as apple scab, apple black rot, grape black rot, and strawberry leaf scorch.

The combined datasets used in this study are summarized in Table 2, and their class distributions are illustrated in Figure 2.

The fruit-disease dataset contains both fruit type and disease labels. It is created by combining the NewPlant Diseases Dataset (Bhattarai, 2018) and the PlantDoc Dataset (Singh *et al*., 2020).

The NewPlant Diseases Dataset is an augmented version of the PlantVillage Dataset, comprising approximately 87,000 RGB images of healthy and diseased crop leaves captured under controlled conditions with uniform backgrounds. It includes 38 classes across 14 plant species. The pre-existing train/test splits are randomly generated, where different augmented versions of the same image may appear in both training and test partitions, leading to data leakage. To prevent this, we restructured the dataset splits by grouping samples according to their original leaf identity. We applied a stratified 80/20 split at the leaf identity level (using a random seed of 42). This ensures that all augmented versions of a given leaf image are assigned exclusively to a single partition (training or test). The PlantDoc dataset consists of 2,598 images collected in real field conditions with varying lighting, occlusions, and complex backgrounds across 13 plant species and up to 17 disease types. The original split of PlantDoc is retained in this work. We merged related species and disease classes to obtain four fruit categories and 11 disease classes relevant to this study. The final dataset contains 21,048 training images and 5,199 test images. Class sizes range from 2,152 to 2,603 images, and the dataset is near-balanced.

The fruit-ripeness dataset contains labels for fruit type and ripeness level. It is created by combining the fruit ripeness level dataset (Diyansah, 2024) and the fruit image dataset (Ahmed, 2023). The fruit ripeness level dataset has seven fruit types at different ripeness levels (unripe, ripe, and overripe), including apple, banana, grape, orange, guava, pomegranate, and strawberry. The fruit image dataset comprises more than 8,700 images of 22 different fruits with ripe and unripe samples.

Before merging, a stratified 80/20 train/test split was applied independently based on fruit type and ripeness label. This ensures that each source contributes proportionally to both partitions. The final dataset contains 7,441 training images and 1,862 test images across four fruit types (apple, grapes, orange, strawberry) and two ripeness levels (ripe, unripe). Class sizes range from 1,048 (strawberry/unripe) to 1,445 (apple/ripe) with ripe/unripe proportions between 50% and 55% for all four fruits (Figure 2).

Each combined dataset contains only two labels per image (fruit + disease or fruit + ripeness), meaning one target label is missing per sample. We handle missing labels during training using a partial-loss masking strategy, where the loss for a missing label is ignored. This allows the backbone to learn from all available images without requiring fully annotated samples.

Data processing

Before training, all images were preprocessed to ensure consistency across the data sources. The preprocessing consists of image resizing, pixel normalization, and data augmentation (applied only during training).

- Image resizing: All images are resized to a fixed resolution of 224×224 pixels, to ensure consistent input for the model.
- Pixel normalization: Images are normalized using the ImageNet mean and standard deviation values [0.485,0.456,0.406] and [0.229,0.224,0.225] respectively, to stabilize training.
- Data augmentation: To improve model generalizability and reduce overfitting, training images are augmented via random horizontal flips and random rotations up to $\pm 30°$.

Backbone model architecture

The MTL model is based on a shared backbone network as shown in Figure 1. The backbone is selected from ResNet-50, EfficientNetV2-M, ViT-B/16, and MobileNetV2 on the basis of their performance trade-offs between accuracy, computational efficiency, and generalizability.

ResNet-50

ResNet-50 (He *et al.*, 2015) is a deep CNN with residual skip connections used to mitigate the problem of vanishing gradients, allowing the training of deep networks. ResNets are constructed by stacking multiple residual bottleneck blocks. ResNet-50 has fifty layers, as shown in Figure 3. The residual block consists of three convolutional layers with batch normalization and ReLU activation, along with a skip connection. The convolutional layer with a kernel size of 3×3 extracts spatial features such as edges, textures, and shapes. The 1×1 convolutional layers at the start and end of the block reduce and then restore the dimensionality of the feature map.

Vision transformer

The vision transformer (ViT) (Dosovitskiy *et al.*, 2021) is a variant of the transformers originally developed for natural language processing but applied directly to images. The image is divided into 16×16 pixel patches, yielding 196 patches in total. Each patch is flattened and converted to a 768-dimensional embedding vector via a linear projection module, as illustrated in Figure 4. These vectors are then sent to a transformer encoder consisting of 12 identical blocks. Each block consists of two main components: a multihead self-attention (MSA) module and a multilayer perceptron (MLP). The MSA mechanism captures global contextual information, enabling the identification of long-range dependencies between image parts. Furthermore, the attention mechanism uses dot products and matrix multiplication that can be performed in parallel, enabling the stacking of multiple layers without slowing down the model.

MobileNetV2

MobileNetV2 (Sandler *et al.*, 2019) is a lightweight, computationally efficient neural network that uses inverted residual blocks and depthwise separable convolutions to significantly reduce the number of parameters and floating-point operations. These properties make it especially suitable for edge or embedded deployment scenarios. The architecture of MobileNetV2 begins with a fully convolutional layer that has 32 filters, followed by 19 residual bottleneck layers, as illustrated in

Table **3**. ReLU6 is used as an activation function because it works well with low-precision calculations. A kernel size of 3 × 3 is commonly used with dropout and batch normalization. The inverted residual block, known as the bottleneck block, is the basic building block. It starts with a 1 × 1 convolution to expand the number of channels. Next, a 3 × 3 depthwise convolution is applied to the expanded feature map, and finally, another 1 × 1 convolution is used to compress the feature map. The depthwise separable convolution divides the convolution process into two simpler steps: depthwise convolution, where a filter is applied separately to each input channel, and a pointwise convolution (1 × 1) that combines these filtered channels to create the final result.

EfficientNet-V2-M

EfficientNet-V2-M (Tan and Le, 2021) belongs to a family of models that use the compound scaling method to uniformly scale network depth, width, and resolution, achieving an effective trade-off between performance and inference speed. EfficientNet-V2 replaces the mobile inverted bottleneck convolution (MBConv) with the fused MBConv (

Table **3**). The MBConv block consists of a 1 × 1 expansion convolution, 3 × 3 depthwise convolution, a squeeze and excitation (SE) block, and 1 × 1 linear projection convolution. The SE module helps the network focus on the most useful features by adjusting the importance of different channels. The Fused MBConv block combines the expansion and depthwise steps into one 3 × 3 convolution to jointly optimize training speed and parameter efficiency.

Convolutional block attention module

To increase the model's ability to focus on critical features, we add a convolutional block attention module (CBAM) (Woo *et al.*, 2018) into the backbone. The CBAM attention mechanism refines feature maps by applying the attention mechanism to two dimensions, channel and spatial, as shown in Figure 5.

Figure 5 The channel attention module (CAM) uses average and max pooling across spatial dimensions, followed by a shared MLP and sigmoid activation to generate channel attention weights. These weights are multiplied element-wise by the feature map to produce the channel feature map. Next, the spatial attention module (SAM) applies average and max pooling along the channel axis, concatenates the results, and applies a convolution layer and sigmoid to generate spatial attention weights. These spatial weights are multiplied by the channel feature map to produce a refined feature map. In this work, we evaluate two configurations that use CBAM. In the first setup, one CBAM module is applied after the shared backbone for all tasks. In the second setup, three separate CBAM layers are used, one for each task, where each attention module can focus on its own output. We compare the two approaches to determine whether focusing attention on each task enhances multitasking performance.

Multitask classification heads

After the backbone and attention modules, three independent task-specific heads are used. Each head consists of two fully connected layers with a ReLU activation function, a dropout layer for regularization, and a final linear layer that maps the features to the corresponding number of output classes. A Softmax activation function is applied to the final output of each head to produce class probabilities.

- Fruit classification: this head identifies the type of fruit (e.g., apple, grape, orange).
- Ripeness classification: this head determines the ripeness stage of the fruit (e.g., ripe, unripe).
- Disease detection: this head classifies whether a disease is present and, if so, its specific type.

Optimization and loss

The model training strategy was designed to effectively optimize the shared and task-specific parameters. The model uses a combined multitask loss (Eq. 1) that jointly optimizes fruit, ripeness, and disease classification.

$$L_{\text{total}} = L_{\text{fruit}} + L_{\text{ripeness}} + L_{\text{disease}}.$$

(Eq. 1)

The loss functions are computed only from available ground-truth labels, allowing the network to learn from annotated data without being penalized for missing annotations. Indeed, for each image i, 2 labels are available, and one is missing. To handle missing annotations, we adopt a partial label masking strategy, also known as masked loss, where a binary mask $m_{i,t}$ indicates whether the label for task t is available for image i ($m_{i,t} = 1$ if available and $m_{i,t} = 0$ otherwise) (Duarte *et al*., 2021). The categorical cross-entropy loss per task t for image i is defined as:

$$L_{i,t} = -\sum_{c=1}^{C_t} m_{i,t}\, y_{i,t,c} \log(\hat{y}_{i,t,c}),$$

(Eq. 2)

where C_t is the number of classes for task t, $y_{i,t,c}$ is the ground-truth label, and $\hat{y}_{i,t,c}$ is the predicted probability for class c. This formulation ensures that for any image where a task label is missing ($m_{i,t} = 0$), no gradient is propagated for that task, effectively skipping the corresponding loss term.

The Adam optimizer was used in two training configurations: a single global optimizer and a decoupled optimization strategy, as shown in Figure 6. In the single-optimizer setup, all parameters were updated using a learning rate of 0.001. In the separate optimizer approach, the backbone and each task-specific head were optimized separately. The backbone was trained with a lower learning rate (0.0001), whereas the task-specific heads used a higher learning rate (0.001) to allow faster adaptation.

Training was conducted using a batch size of 64 for 20 epochs on an NVIDIA GPU-based platform with PyTorch. To ensure reproducibility and account for variability due to random initialization and data shuffling, all experiments were repeated three times using different random seeds (42, 123, 456); the mean and standard deviation of the performance across these runs are reported. In addition, k-fold (k=5) cross-validation was performed using the MobileNetV2 backbone to provide a more robust and unbiased evaluation of the model's generalization performance.

Performance metrics

The performance of the proposed MTL model is evaluated using standard multiclass metrics, including accuracy, precision, recall, and the F1-score. Model behavior is further analyzed through confusion matrices for each task. The macro-averaged metrics of all m classes are reported.

The accuracy measures the overall proportion of correct predictions made by the model. It is computed as:

$$\text{Accuracy} = \frac{\sum_{i \in m} (\text{TP}_i + \text{TN}_i)}{\sum_{i \in m} (\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i)}, \tag{Eq. 3}$$

The precision for the i^{th} class measures the correctness of positive predictions. It is defined as:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \qquad \text{Precision}_{macro} = \frac{1}{m} \sum_{i=1}^{m} \text{Precision}_i\,, \tag{Eq. 4}$$

The recall for the i^{th} class measures the model's ability to correctly identify all samples belonging to that class:

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \qquad \text{Recall}_{macro} = \frac{1}{m} \sum_{i=1}^{m} \text{Recall}_i\,. \tag{Eq. 5}$$

The F1-score is the harmonic mean of precision and recall:

$$\begin{aligned} \text{F1-score}_i &= \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}, \\ \text{F1-score} &= \frac{1}{m} \sum_{i=1}^{m} \text{F1-score}_i\,. \end{aligned} \tag{Eq. 6}$$

The evaluation focuses on both accuracy and the macro-averaged F1-score, which provides a balanced measure of precision and recall across all classes.

Results and Discussion

This section presents the performance of the proposed MTL framework for joint fruit, ripeness, and disease classification. The model is analyzed under different backbone architectures, CBAM configurations, and optimization strategies. Additionally, per-source and cross-dataset evaluation experiments are conducted to assess generalization across domains.

Backbone comparison

To evaluate the impact of feature extraction on MTL performance, the proposed framework was trained using four backbones: ResNet-50, ViT-B/16, EfficientNetV2-M, and MobileNetV2. All the networks were trained for 20 epochs via a single global optimizer and without CBAM. The results are reported in Table 4. All values were reported as mean (%) over three runs, with all standard deviations below 0.5%, indicating strong training stability across all tasks and backbones.

In all backbone architectures, disease classification achieved consistently high performance, exceeding 99% accuracy. Fruit type classification also achieved strong performance with EfficientNetV2-M and MobileNetV2 achieving respectively an accuracy of 98.95% (F1: 98.89%) and 98.63% (F1: 98.5%). These findings indicate that fruit type and disease classification relied on discriminative visual patterns, such as shape, texture anomalies, and lesion patterns, that could be captured effectively even by lightweight backbones. In contrast, ripeness estimation presented lower performance because the differences between ripe and unripe fruits are more subtle, involving changes in color and texture that are harder to distinguish. All the models achieved accuracies above 90%, with EfficientNetV2-M having the highest accuracy (96.12%), closely followed by MobileNetV2 (95.84%). ViT-B/16 presented slightly lower performance, with 93% for ripeness and 92.44% for fruit type. These results show that EfficientNetV2-M is the best performing backbone. However, MobileNetV2, despite its significantly lower computational cost (Table 5) performed competitively, achieving accuracies of 98.63%, 95.84%, and 99.43% for fruit, ripeness, and disease, respectively, only marginally below EfficientNetV2-M, demonstrating its suitability for deployment in resource-constrained settings. Table 5 compares the computational costs of the four backbones, including model size, inference time, parameter count, and floating-point operations (FLOPs). All

measurements were obtained under the same hardware configuration (Ubuntu 20.04, 24 GB RAM, 8 CPU cores, and an NVIDIA RTX A4000 GPU).
MobileNetV2 demonstrated the highest computational efficiency, requiring only 5.48 M parameters and 2.61 GFLOPs, and achieved the fastest inference time of 6.75 ms. ResNet-50 also presented moderate computational costs with 28.7 M parameters, 33.06 GFLOPs, and an inference time of 8.13 ms. ViT-B/16, despite its transformer-based design that enables parallel attention operations, remained the largest model in terms of parameter count (86.99 M) and model size (331.9 MB), with a slightly slower inference time of 8.51 ms. EfficientNetV2-M presented the slowest inference time of 25.29 ms with over 56 M parameters and 43.57 GFLOPs. These results demonstrate that lightweight models provide the best tradeoff between accuracy and efficiency, making them particularly attractive for real-time agricultural monitoring systems.

Effect of optimizer strategies

To study the impact of the optimization strategy on multitask learning, two training strategies were compared, as previously discussed. The first approach used a single global optimizer for all the parameters, whereas the second approach assigned separate optimizers to the backbone and each task-specific head. Table 6 shows the results across ResNet-50, ViT-B/16, and MobileNetV2 backbones.
Overall, the separate-optimizer strategy consistently improved performance across all backbones and tasks, though the magnitude of improvement varied. The most significant gains were observed with the ViT-B/16 backbone, where fruit accuracy increased from 92.44% to 99.32%, with the F1-score improving from 91.97% to 99.22%. Similarly, ripeness classification accuracy improved from 93.00% to 96.15%, with the F1-score increasing from 92.99% to 96.14%, indicating a substantial benefit from task-specific optimization.
For ResNet-50 and MobileNetV2, improvements were smaller but consistent. For ResNet-50, fruit classification increased from 98.49% to 99.40% accuracy (F1: 98.34% to 99.33%), ripeness from 95.81% to 96.04%, and disease from 99.42% to 99.67%. For MobileNetV2, fruit improved from 98.63% to 99.29% accuracy (F1: 98.50% to 99.24%), ripeness from 95.84% to 95.95%, and disease from 99.43% to 99.65%.
The reported standard deviations remained low across all configurations (<0.5%), indicating stable training behavior. Additionally, the improvements in accuracy were consistently aligned

with gains in macro-averaged F1-scores, confirming that the observed performance improvements were balanced across classes.

These results highlight the advantages of using task-specific optimization. This strategy enables each task to learn more specialized representations from its task-specific learning rates and gradient flows.

Impact of CBAM

To evaluate the influence of attention mechanisms on feature refinement, CBAM modules were added to the MobileNetV2 model in two configurations: a single shared CBAM after the backbone, and three CBAM modules applied individually to each task-specific head. Both configurations were trained via separate optimizers. The corresponding results are summarized in Table 7.

Using attention significantly improved the model's ability to focus on task-relevant features. Adding a single CBAM module improved performance across all tasks. For example, the ripeness F1-score increased from 95.95% to 96.34%, with accuracy improving from 95.95% to 96.35%. Fruit and disease classification also showed marginal gains, reaching 99.36% F1 / 99.39% accuracy and 99.69% F1 / 99.69% accuracy, respectively.

In contrast, the three-CBAM configuration did not provide further benefits. Although fruit and disease classification showed slight improvements (fruit: F1 99.48% / Acc 99.51%; disease: F1 99.75% / Acc 99.75%), ripeness performance dropped back to 95.95%. This can be explained by over-refinement or redundant attention that does not generalize well.

K-fold cross-validation (K=5) was conducted to assess the generalizability and stability of the observed performance trends beyond a single train/test split, as reported in Table S4. As expected, the k-fold estimates are slightly lower than those in Table 7, as performance is averaged over five different splits. The results largely confirmed the superiority of the single-CBAM configuration for fruit and disease classification. However, the 3-CBAM configuration achieved the highest ripeness F1-score (96.55%), suggesting that the ripeness degradation observed in Table 7 may be sensitive to the train/test split.

Overall, a single shared CBAM module provides the best trade-off between performance and model complexity. This CBAM layer refines feature maps by applying both channel and spatial attention, which helps the model weigh the importance of different features and regions of an image.

The confusion matrices for the MTL model (MobileNetV2 with one CBAM and separate optimizers) are presented in Figure 7. These visualizations provide deeper insights into the per-class performance in the three prediction tasks. For fruit type classification, the model correctly identifies each fruit category with an accuracy of 99%. Misclassification rates remain below 0.8% and occur primarily between oranges, apples, and grapes, likely owing to similarities in green color. For ripeness, the model correctly identifies unripe samples with an accuracy of 98.2%, whereas ripe samples present a slightly higher misclassification rate of 5.6%. These findings support earlier observations that ripeness presents a more challenging problem. The disease confusion matrix demonstrated near-perfect class separation across all disease categories. Most samples are classified with 99% accuracy, with only a minor confusion observed between apple diseases and apple and grapes at 0.4%.

Interpretability using Grad-CAM

To better understand and interpret the predictions of the MTL model, we use gradient-weighted class activation mapping (Grad-CAM) as an explainable AI (XAI) tool. Grad-CAM generates heatmaps that highlight the important image regions that the model uses when making a specific class prediction. This helps assess whether the model focuses on relevant areas when performing fruit type, ripeness, and disease classification. Grad-CAM uses feature maps from the last conv layer to compute the gradients of the model's output with respect to these maps. These gradients indicate how important each feature map is for the target class. A global average pooling operation is then applied to obtain a single importance weight per feature map, and a weighted combination of all feature maps is passed through ReLU activation to produce a class activation map.

For the MTL model using a MobileNetV2 backbone without CBAM, the 17th inverted residual block is selected as the target convolutional layer, as it has high-level, semantically rich features while retaining some spatial information. For the MobileNetV2 backbone with CBAM, the CBAM module itself is used as the target layer, allowing visualization of refined feature maps. This enables direct observation of how the CBAM module influences the model's focus for each task-specific prediction.

Figure **8**a shows the Grad-CAM visualizations for the MTL model using a MobileNetV2 backbone, with and without the CBAM attention module, for an apple image. The model without CBAM misclassifies the fruit type and disease. In contrast, the model with CBAM focuses more clearly on the fruit regions, resulting in correct fruit classification.

Figure **8**b presents another example using an orange image and illustrates how CBAM improves ripeness-related attention. Without CBAM the model produces diffuse attention outside the fruit region and an incorrect ripeness prediction. When CBAM is applied, the attention becomes more compact and centered on the fruit, leading to correct ripeness classification and improved disease localization. Although the disease label remains incorrect, the model focuses on a more appropriate region for orange disease. It is important to note that no healthy class was available for oranges during training.

In the example of the grape leaf (Figure 8c), both models correctly classified the leaf as a grape. However, the model without CBAM fails in disease classification, whereas the CBAM model shows more concentrated attention on the lesion regions, leading to a correct disease prediction. Regarding ripeness, this image does not contain a ripeness label, since ripeness applies to fruits rather than leaves, which explains the inconsistent heatmaps observed for this task.

In summary, Grad-CAM visualizations demonstrate that CBAM helps the model focus on task-relevant regions—such as fruit surfaces and disease lesions—leading to more reliable and realistic decisions.

Multitask *vs* single task model performance

To assess the benefits of multitask learning, we compared the final MTL configuration (MobileNetV2 with a shared CBAM module and separate optimizers) against three independently trained single-task MobileNetV2 models. Each single-task model included the same CBAM module. The results are summarized in Table S1.

The MTL model outperformed the single-task models in fruit classification, increasing accuracy from 98.05% to 99.39% (F1: 98.07% to 99.36%). A marginal improvement was observed for ripeness prediction, where the accuracy increased from 96.06% to 96.35% (F1: 96.06% to 96.34%). Disease classification showed strong performance in both cases, with the MTL model achieving 99.69% accuracy compared to 99.56% for single-task models.

K-fold cross-validation (K=5) further confirmed these findings. As expected, the k-fold estimates were slightly lower than the train/test split results. The MTL model consistently outperformed the single-task model across all tasks, achieving a fruit F1-score of 98.31% *vs* 97.75%, ripeness F1-score of 96.33% *vs* 96.04%, and disease F1-score of 99.63% *vs* 99.27%, confirming the generalizability and stability of the MTL approach.

This outcome demonstrates that by sharing a common backbone, the model learns a richer feature representation that benefits all tasks. Moreover, this approach reduces the computational cost by replacing three independent models with a single unified network, lowering the memory and inference requirements.

Cross-dataset evaluation

To assess the generalization capability of the proposed model, two complementary evaluations were conducted: a per-source evaluation (Table S2) and a cross-dataset evaluation (Table S3). In the cross-dataset setting, the model was trained on the NewPlant and Fruit Ripeness Level datasets and evaluated on unseen domains (PlantDoc and Fruit Image-22 Classes).

In the per-source evaluation, the model achieved near-perfect scores (fruit F1: 99.98%, disease F1: 99.93%) on NewPlant, a controlled dataset with uniform backgrounds. Performance on PlantDoc was lower (fruit F1: 89.89%, disease F1: 79.24%), which was expected given that PlantDoc consists of real-world field images with complex backgrounds, variable lighting, and greater noise. The small subset size (57 validation images) further increased variance. For Fruit Ripeness Level subset, the model achieved strong performance (fruit F1: 99.32%, ripeness F1: 99.25%). In Fruit Image (22 Classes), the model showed lower results (fruit F1: 93.35%, ripeness F1: 85.62%) due to its smaller size and noisier images.

The cross-dataset evaluation further highlighted the impact of domain shift. Disease classification dropped substantially from 99.93% (NewPlant) to 59.68% F1 (PlantDoc), demonstrating limited robustness when trained without field-condition data. When PlantDoc was included during training (Table S2) disease performance improved significantly to 79.24%, indicating the importance of domain-specific training data. In contrast, fruit classification generalized well across datasets, with only a minor decrease, suggesting that fruit-type features such as color and shape are more transferable across domains.

These results demonstrated that performance variation across sources was driven primarily by image acquisition conditions and dataset complexity. The inclusion of more diverse and realistic training data could further address these limitations in future work.

Summary of results

Overall, the experiments highlight the following:

- MobileNetV2 with a single CBAM trained with separate optimizers represents the best trade-off between accuracy and computational efficiency.
- The MTL model consistently outperforms single-task models.

- Adding attention mechanisms, one CBAM module after the shared backbone, significantly improves feature refinement.
- Grad-CAM explainability provides insight into the model's focus and decision-making.
- Disease detection was most sensitive to domain shift, underscoring the need for more realistic training data.

Conclusions and future work

This work presents an efficient MTL model for joint fruit, ripeness, and disease classification using a single lightweight MobileNetV2 backbone enhanced with a CBAM. By integrating three related tasks into one unified architecture, the proposed system reduces computational redundancy, improves feature sharing across tasks, and enables real-time deployment. Combined datasets were constructed using a label-masking loss strategy to handle missing annotations during training. We also compared several backbone networks, including ResNet-50, EfficientNetV2-M, and ViT-B/16, alongside different optimization strategies and CBAM configurations.

The experimental results demonstrate that using separate optimizers leads to faster convergence and better generalizability. In addition, MTL models achieved higher accuracy compared to independent single-task models. In particular, the MobileNetV2 model with a single shared CBAM module achieved 99.39% fruit accuracy, 99.69% disease accuracy, and 96.35% ripeness accuracy, while having the lowest parameter count and FLOPs among all evaluated backbones. These results confirm that lightweight shared models with attention mechanisms can effectively capture fine-grained visual cues such as texture, color, and disease symptoms. The combination of high accuracy, fast inference, and low resource consumption makes this approach suitable for real-time deployment on edge devices.

Future research directions include expanding the model to support additional categories of fruits and diseases, incorporating more diverse real-field images, and enabling scalability to broader agricultural environments. Lightweight transformer-based architectures (Hu *et al*., 2025) could be further investigated to explore additional gains, particularly for ripeness. Integrating disease localization into the joint model is another promising direction. Finally, semi-supervised and self-supervised learning methods could be used to leverage large amounts of unlabeled field data, helping to address the limited availability of fully annotated agricultural datasets.

Table S1. Summary of accuracy.
Table S2. Per-source evaluation.
Table S3. Cross-dataset generalization.
Table S4. Summary of performance across Tasks and CBAM configurations.

References

Ahmed MS, 2023. Fruit Image Dataset: 22 Classes. Kaggle. Available from: https://www.kaggle.com/datasets/mdsagorahmed/fruit-image-dataset-22-classes

Ali A, 2019. PlantVillage Dataset. Kaggle. Available from: https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

Atila Ü, Uçar M, Akyol K, Uçar E, 2021. Plant leaf disease classification using EfficientNet deep learning model. Ecol Inform 61:101182.

Attri I, Awasthi LK, Sharma TP, Rathee P, 2023. A review of deep learning techniques used in agriculture. Ecol Inform 77:102217.

Bai C, Zhang L, Gao L, Peng L, Li P, Yang L, 2024. DINOV2-FCS: a model for fruit leaf disease classification and severity prediction. Front Plant Sci 15:1475282.

Bhattarai S, 2018. New Plant Diseases Dataset. Kaggle. Available from: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

Dai G, Tian Z, Fan J, Sunil CK, Dewi C, 2024. DFN-PSAN: multi-level deep information feature fusion extraction network for interpretable plant disease classification. Comput Electron Agric 216:108481.

Diyansah D, 2024. Fruit Ripeness Level Dataset. Kaggle. Available from: https://www.kaggle.com/datasets/dudinurdiyansah/fruit-ripeness-level-dataset/data

Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al., 2021. An image is worth 16×16 words: transformers for image recognition at scale. arXiv:2010.11929.

Duarte K, Rawat YS, Shah M, 2021. PLM: partial label masking for imbalanced multi-label classification. arXiv:2105.10782.

Fu R, Wang X, Wang S, Sun H, 2025. PMJDM: a multi-task joint detection model for plant disease identification. Front Plant Sci 16:1599671.

Hamed BS, Hussein MM, Mousa AM, 2023. Plant disease detection using deep learning. Int J Intell Syst Appl 15:38-50.

He K, Zhang X, Ren S, Sun J, 2015. Deep residual learning for image recognition. arXiv:1512.03385.

Hemalatha S, Jayachandran JJB, 2024. A multitask learning-based vision transformer for plant disease localization and classification. Int J Comput Intell Syst 17:188.

Hu D, Zhao L, Ren Y, Wang S, Ye X, Zhang H, et al., 2025. A lightweight vision transformer with embedded hybrid attention for quick response code defect classification. Eng Appl Artif Intell 150:110575.

Krishna MS, Machado P, Otuka RI, Yahaya SW, Neves dos Santos F, Ihianle IK, 2025. Plant leaf disease detection using deep learning: a multi-dataset approach. J 8:4.

Mi Z, Yan WQ, 2024. Strawberry ripeness detection using deep learning models. Big Data Cogn Comput 8:92.

Nair S, Bajpai K, Fadnavis S, Jain S, Sheikh Z, 2024. Plant Fettle Detector: unifying fruit quality tracking and plant disease detection through advanced image analysis. In: Proceedings of the 2024 3rd International Conference for Innovation in Technology (INOCON). IEEE; pp. 1-5.

Padhiary M, Saha D, Kumar R, Sethi LN, Kumar A, 2024. Enhancing precision agriculture: a comprehensive review of machine learning and AI vision applications in all-terrain vehicles for farm automation. Smart Agric Technol 8:100483.

Sandhi A, 2024. A comprehensive hybrid model for apple fruit disease detection using multi-architecture feature extraction. Preprints 202410.1894.v1.

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC, 2019. MobileNetV2: inverted residuals and linear bottlenecks. arXiv:1801.04381.

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D, 2017. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). IEEE; pp. 618-26.

Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N, 2020. PlantDoc: a dataset for visual plant disease detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (CoDS COMAD 2020). Association for Computing Machinery; pp. 249-53.

sshikamaru, 2018. Fruit Recognition. Kaggle. Available from: https://www.kaggle.com/datasets/sshikamaru/fruit-recognition

Sultana N, Jahan M, Uddin MS, 2022. An extensive dataset for successful recognition of fresh and rotten fruits. Data Brief 44:108552.

Tan M, Le QV, 2021. EfficientNetV2: smaller models and faster training. arXiv:2104.00298.

Tapia-Mendez E, Cruz-Albarran IA, Tovar-Arriaga S, Morales-Hernandez LA, 2023. Deep learning-based method for classification and ripeness assessment of fruits and vegetables. Appl Sci 13:12504.

Uddin AH, 2020. Plant Doc Dataset. Kaggle. Available from: https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset

Wang CY, Yeh IH, Liao HYM, 2024. YOLOv9: learning what you want to learn using programmable gradient information. arXiv:2402.13616.

Wang P, Niu T, Mao Y, Zhang Z, Liu B, He D, 2021. Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism. Front Plant Sci 12:723294.

Woo S, Park J, Lee JY, Kweon IS, 2018. CBAM: convolutional block attention module. arXiv:1807.06521.

Yang B, Wang Z, Guo J, Guo L, Liang Q, Zeng Q, et al., 2023. Identifying plant disease and severity from leaves: a deep multitask learning framework using triple-branch Swin Transformer and deep supervision. Comput Electron Agric 209:107809.

Yu J, Dai Y, Liu X, Huang J, Shen Y, Zhang K, et al., 2024. Unleashing the power of multi-task learning: a comprehensive survey spanning traditional, deep, and pretrained foundation model eras. arXiv:2404.18961.

Zhang W, 2023. A fruit ripeness detection method using adapted deep learning-based approach. Int J Adv Comput Sci Appl 14:121.
Zhang Y, Yang X, Cheng Y, Wu X, Sun X, Hou R, et al., 2024. Fruit freshness detection based on multi-task convolutional neural network. Curr Res Food Sci 8:100733.

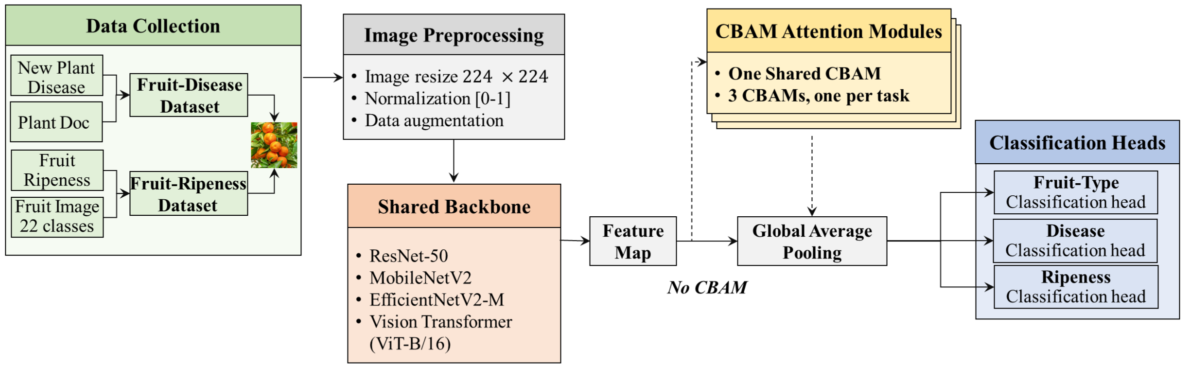

Figure 1. Proposed MTL model for fruit type, ripeness, and disease classification using a shared backbone (ResNet-50, MobileNetV2, EfficientNet, ViT) with three CBAM setups: none, shared, or task specific.

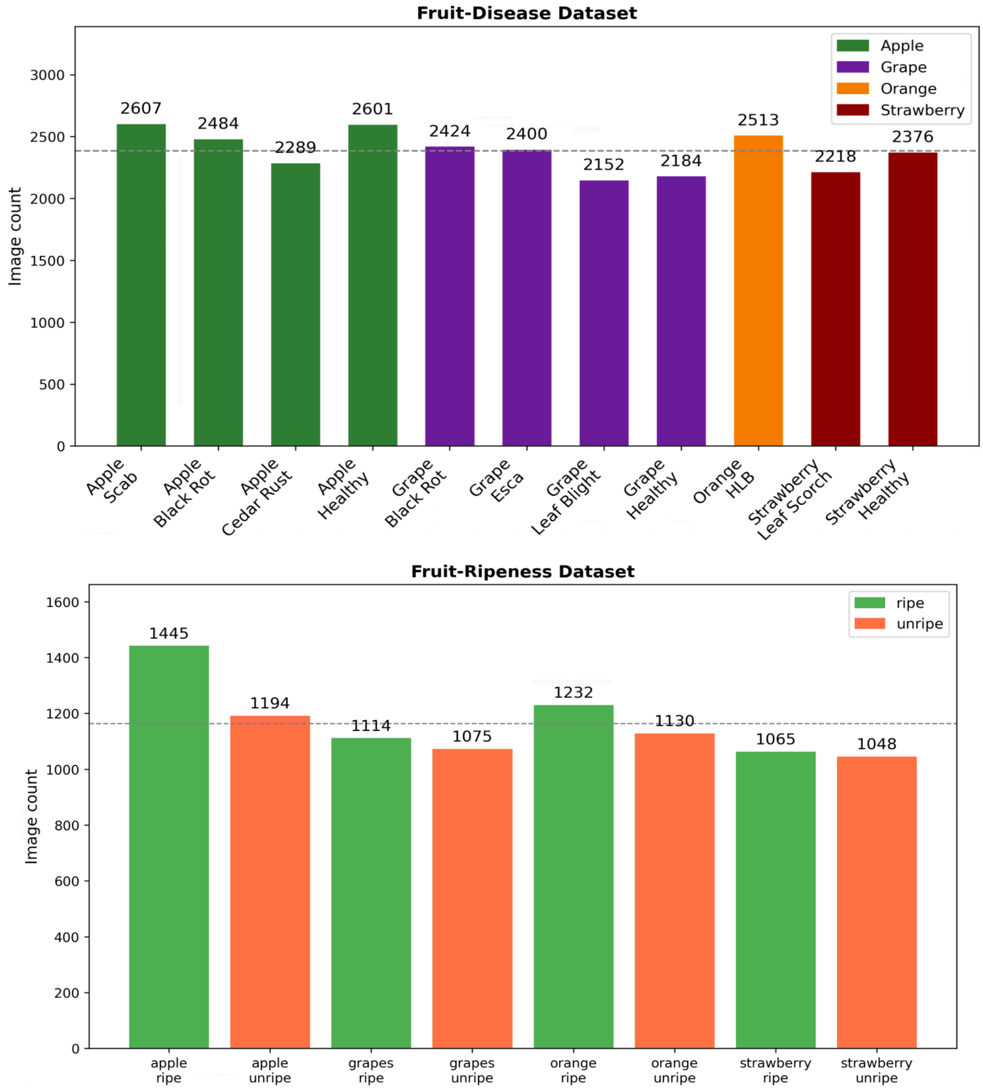

Figure 2. Class distribution histograms for the fruit–disease (top) and fruit–ripeness (bottom) datasets, showing generally balanced sample counts across classes.

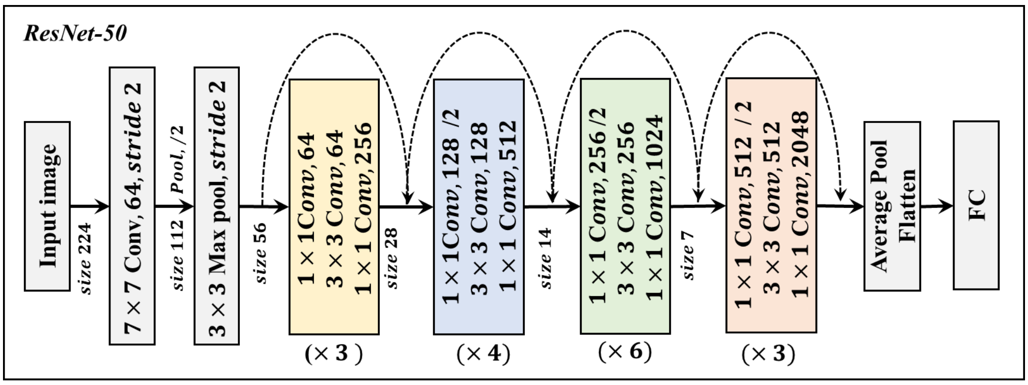

Figure 3. ResNet-50 architecture with stacked residual bottleneck blocks and skip connections.

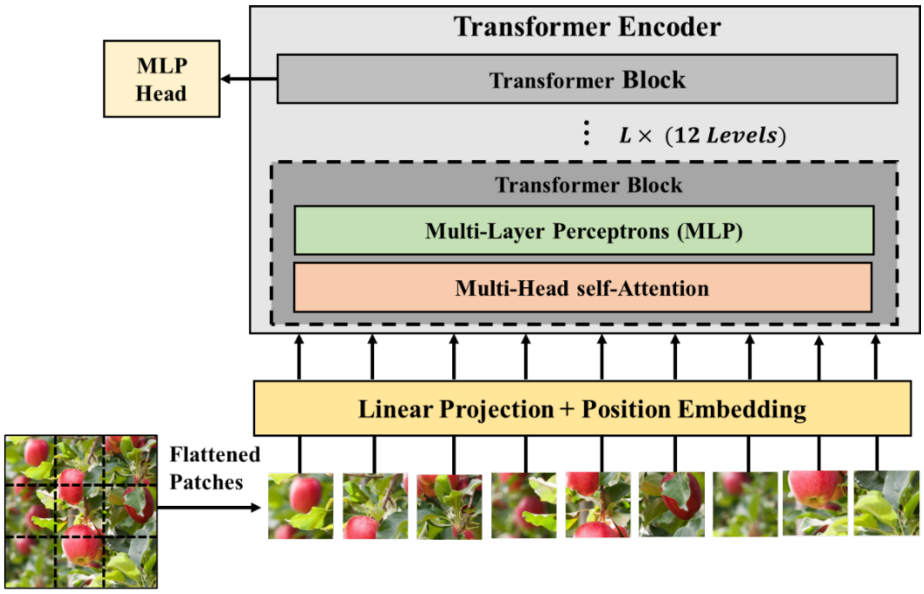

Figure 4. Vision transformer (ViT) architecture, showing patch embedding and stacked transformer encoder blocks.

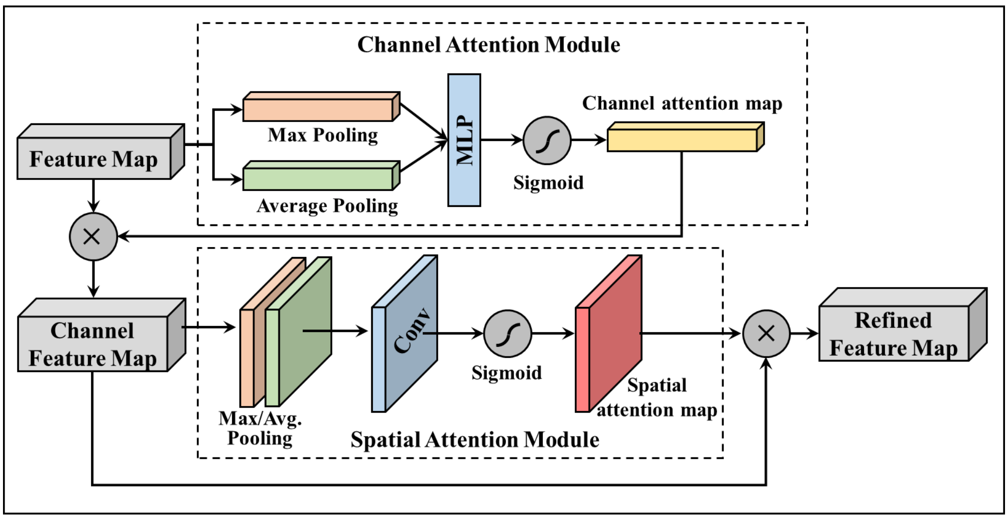

Figure 5. Convolutional block attention module (CBAM) with sequential channel and spatial attention.

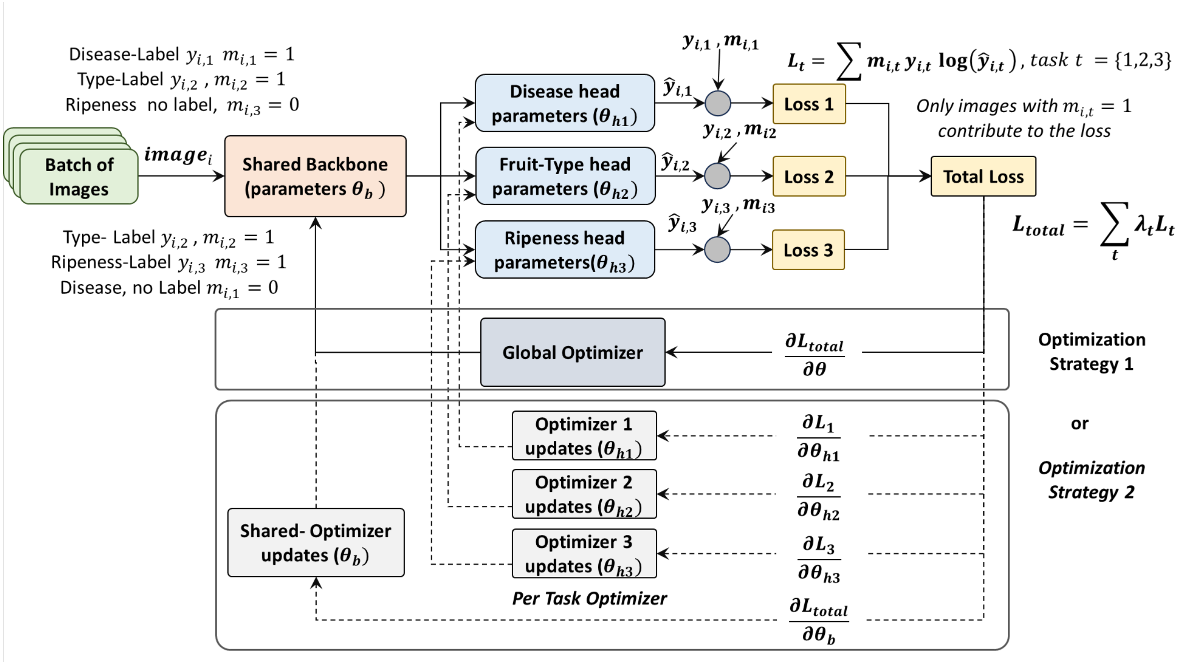

Figure 6. MTL training process with partial label masking, and the use of either global or task-specific optimizers.

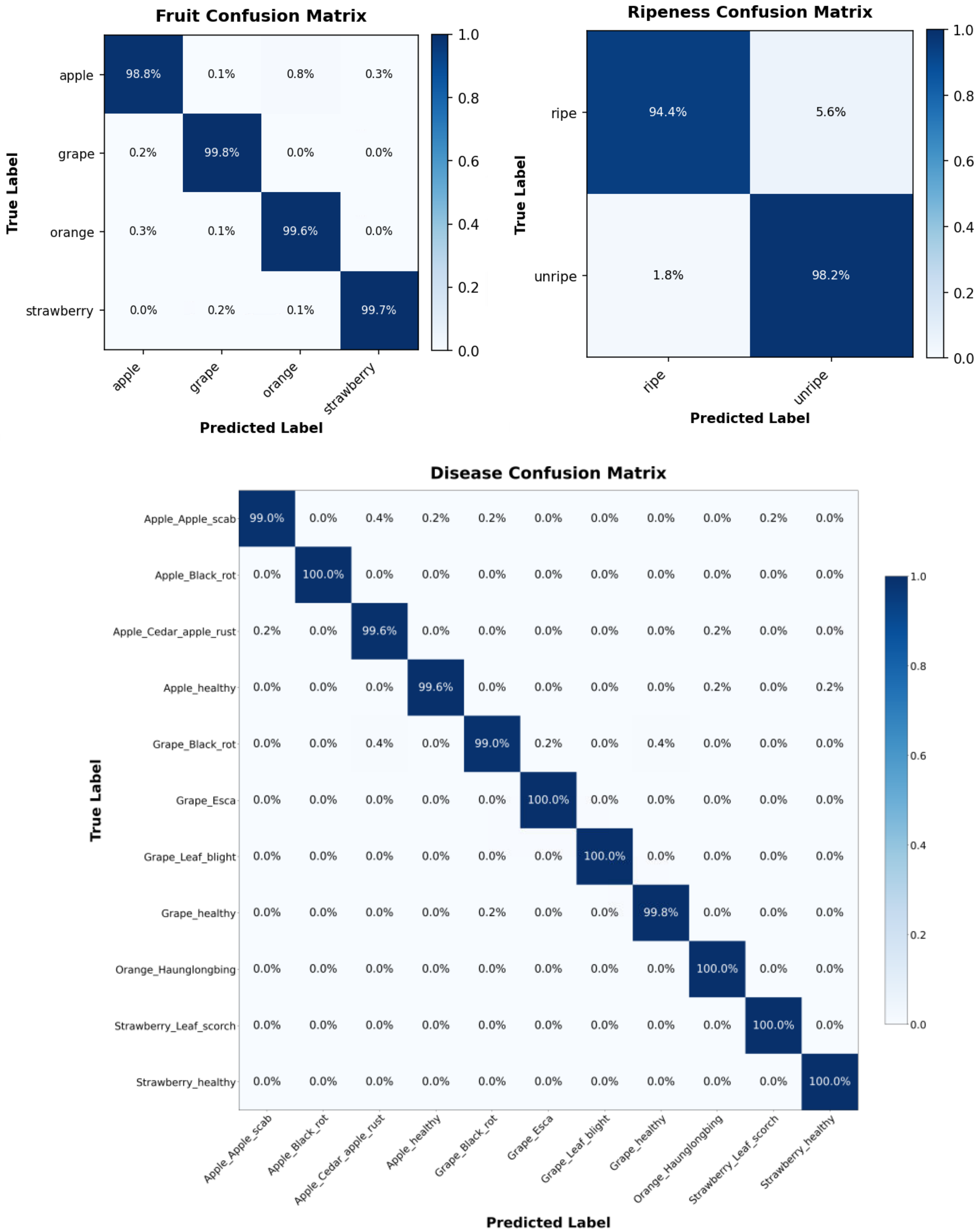

Figure 7. Normalized confusion matrices for (a) fruit type classification, (b) ripeness classification, (c) disease classification.

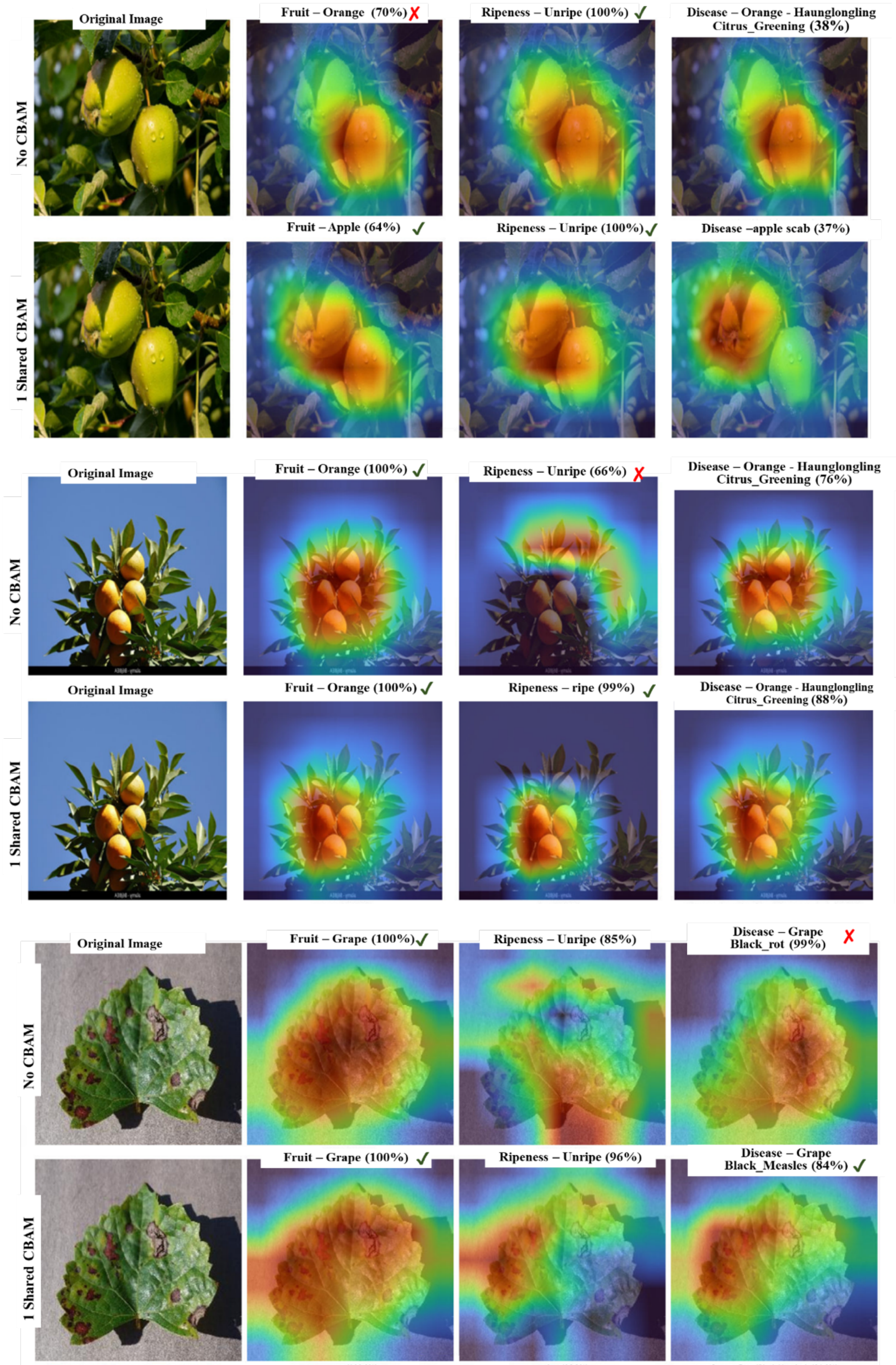

Figure 8. Grad-CAM visualizations of the MTL MobileNetV2 model without CBAM (top) and with CBAM (bottom) for (a) apple, (b) orange, and (c) grape leaf images. CBAM improves the focus on task-relevant regions across all three tasks.

Table 1. Summary of recent DL models for plant disease, ripeness, and fruit classification tasks.

Reference	Dataset	Model	Task	Results
(Atila *et al.*, 2021)	PlantVillage	EfficientNet B5, B4	Disease classification	Acc. 99.91–99.97%
(Wang *et al.*, 2021)	Apple Leave (ALDID) from PlantVillage	CA-ENet (Attention+ EfficientNet)	Apple leaf disease classification	Acc. 98.92%
(Hamed *et al.*, 2023)	NewPlant Diseases (38 classes)	EfficientNetV2S	Disease detection under field-like noise	Acc. 95.01%
(Yang *et al.*, 2023)	AI Challenger 2018 dataset	Swin Transformer	Disease and severity classification	Acc. 99.00% (disease), 88.73% (severity)
(Dai *et al.*, 2024)	Katra-Twelve, Sunflower, FGVC8	DFN-PSAN	Disease classification	Acc. 95.27%, F1 95.27%
(Bai *et al.*, 2024)	PlantVillage (5 fruits)	DINOV2-FCS (Transformer + C-PFFM)	Disease classification and lesion segmentation	Acc. 99.67%
(Krishna *et al.*, 2025)	PlantDoc and web-sourced	EfficientNet-B3 (ResNet, DenseNet)	Disease classification	Acc. 80.19%, F1 90%
(Tapia-Mendez *et al.*, 2023)	Fruit Classification	Dual MobileNetV2	Fruit type (32 classes) and ripeness classification	Acc. 97.86% (fruit), 100% (ripeness)
(Zhang, 2023)	Custom multi-fruit	YOLOv8	Ripeness detection	Acc. 98%
(Mi and Yan, 2024)	Strawberry Dataset	YOLOv9 + Swin Transformer	Ripeness detection	mAP 87.3%
(Zhang *et al.*, 2024)	Fresh and rotten fruits dataset	MTL: Shared CNN + 2 FC heads	Freshness + fruit-type classification	Acc. 93.24% (freshness), 88.66% (fruit)
(Hemalatha and Jayachandran, 2024)	PlantVillage, PlantDoc	MTL: PDLC-ViT	Disease detection + localization	Acc. 99.97%, mAP 99.18%
(Fu *et al.*, 2025)	Multiple plant datasets	MTL: ConvNeXt	Species + Disease localization	mAP50 61.83%

Table 2. Summary of the combined datasets used for fruit type, ripeness, and disease classification, including source datasets, available labels, and sample distributions.

Combined Dataset	Source Datasets	Labels available	Number of Samples
Fruit-disease dataset	NewPlant Diseases (Bhattarai, 2018) PlantDoc (Singh *et al.*, 2020)	Fruit type (4 types) Disease (11 Diseases)	Train: 21,030 Test: 5,198
Fruit-ripeness dataset	Fruit Ripeness Level (Diyansah, 2024) Fruit Image-22 Classes (Ahmed, 2023)	Fruit type (4 types) Ripeness (2 types)	Train: 7,441 Test: 1,862

Note: The four fruit types are: apple, grape, orange, and strawberry.
The ripeness types include ripe or unripe.
The disease types: apple scab, apple black rot, cedar apple rust, healthy apple, grape black rot, grape black measles, grape leaf blight, healthy grape, orange citrus greening, strawberry leaf scorch, and healthy strawberry.

Table 3. Backbone architectures: MobileNetV2 and EfficientNetV2-M.

MobileNetV2 architecture

Layer	Input Size	#Channels	Stride	Expansion
Conv2D	224×224 × 3	32	2	–
bottleneck	112×112 × 32	16	1	1
bottleneck	112×112 × 16	24	2	6
bottleneck	56×56 × 24	32	2	6
bottleneck	28×28 × 32	64	2	6
bottleneck	14×14 × 64	96	1	6
bottleneck	14×14 × 96	160	2	6
bottleneck	7×7 × 160	320	1	6
Conv2D 1×1	7×7 × 320	1280	–	–
AvgPool 7×7	7×7 × 1280	–	–	–
Conv2D 1×1	1×1 × 1280	C	–	–

EfficientNetV2-M architecture

Stage	Block Type	#Layers	#Channels	Stride
0	Conv 3×3	1	24	2
1	Fused-MBConv	2	24	1
2	Fused-MBConv	4	48	2
3	Fused-MBConv	4	64	2
4	MBConv-SE	6	128	2
5	MBConv-SE	9	160	1
6	MBConv-SE	15	256	2
7	Conv 1×1	1	1280	1

Table 4. Summary of MTL model performance using multiple backbones across all tasks, evaluated under the No-CBAM configuration using a single optimizer.

Backbone	Fruit				Ripeness				Disease			
	Prec	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**
ResNet-50	98.21	98.48	98.34	98.49	95.85	95.77	95.80	95.81	99.42	99.43	99.42	99.42
ViT-B/16	92.20	91.78	91.97	92.44	93.01	92.99	92.99	93.00	97.34	97.32	97.32	97.30
MobileNetV2	98.48	98.53	98.50	98.63	95.83	95.86	95.84	95.84	99.43	99.44	99.43	99.43
EfficientNetV2-M	**98.91**	**98.87**	**98.89**	**98.95**	**96.11**	**96.12**	**96.11**	**96.12**	**99.65**	**99.66**	**99.65**	**99.65**

All values are reported as mean (%) over three runs. Standard deviations for F1/Acc
ResNet-50: Fruit (±0.32/±0.27), Ripe (±0.23/±0.23), Disease (±0.03/±0.10);
*ViT-B/16: Fruit (**±0.42/±0.37**), Ripe (±0.35/±0.35), Disease (**±0.25/±0.25**);*
*MobileNetV2: Fruit (±0.18/±0.16), Ripe (**±0.37/±0.37**), Disease (±0.06/±0.05);*
EfficientNetV2-M: Fruit (±0.04/±0.04), Ripe (±0.08/±0.08), Disease (±0.04/±0.03)

Table 5. Comparison of computational efficiency across backbones, including inference time, model size, parameter count, and FLOPs.

Backbone	Inference Time (ms)	Model Size (MB)	Total Parameters	GFlops
ResNet-50	8.13	109.9	28,713,017	33.06
ViT-B/16	8.51	331.9	86,988,561	22.57
MobileNetV2	**6.75**	**21.2**	**5,481,209**	**2.61**
EfficientNetV2-M	25.29	215.8	56,115,693	43.57

Table 6. Summary of accuracy and F1 scores across tasks for different optimizer strategies.

Backbone	Optimizer Strategy	Fruit		Ripeness		Disease	
		F1	Acc	F1	Acc	F1	Acc
ResNet	Single Global	98.34	98.49	95.80	95.81	99.42	99.42
	Separate Optimizers	**99.33**	**99.40**	96.04	96.04	**99.67**	**99.67**
MobileNetV2	Single Global	98.50	98.63	95.84	95.84	99.43	99.43
	Separate Optimizers	99.24	99.29	95.95	95.95	99.66	99.65
ViT	Single Global	91.97	92.44	92.99	93.00	97.32	97.30
	Separate Optimizers	99.22	99.32	**96.14**	**96.15**	99.63	99.63

Standard deviations (F1/Acc):
*ResNet-50 3-opt: Fruit **(±0.17/±0.16)**, Ripe (**±0.47/±0.47**), Disease (**±0.12/±0.12**);*
MobileNetV2 3-opt: Fruit (±0.06/±0.07), Ripe (±0.46/±0.45), Disease (±0.04/±0.04);
ViT-B/16 3-opt: Fruit (±0.13/±0.11), Ripe (±0.25/±0.24), Disease (±0.09/±0.09).

Table 7. Summary of Accuracy Across Tasks and CBAM Configurations using Separate Optimizers.

Backbone	Fruit				Ripeness				Disease			
	Prec	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**
MobileNetV2 No-CBAM	99.24	99.24	99.24	99.29	95.95	96.01	95.95	95.95	99.66	99.72	99.66	99.65
MobileNetV2 1-CBAM	99.29	99.43	99.36	99.39	**96.33**	**96.38**	**96.34**	**96.35**	99.70	99.70	99.69	99.69
MobileNetV2 3-CBAM	**99.44**	**99.53**	**99.48**	**99.51**	95.93	95.99	95.95	95.95	**99.75**	**99.75**	**99.75**	**99.75**
Standard deviations (F1/Acc): *No-CBAM: Fruit (±0.06/±0.07), Ripe (**±0.45/±0.45**), Disease (±0.04/±0.04);* *1-CBAM: Fruit (±0.06/±0.05), Ripe (±0.19/±0.18), Disease (**±0.09/±0.09**);* *3-CBAM: Fruit (**±0.12/±0.13**), Ripe (±0.34/±0.35), Disease (±0.04/±0.04).*												