A Faster RCNN Architecture for Simultaneous Detection of Fruit type and Disease : A Multi-task Learning Approach

Original Scientific Paper

Seema Shrawne*
Veermata Jijabai Technological Institute,
Department of Computer Engineering and Information Technology,
Matunga, Mumbai, India
scshrawne@ce.vjti.ac.in

Kaustubh Chile
Veermata Jijabai Technological Institute,
Department of Computer Engineering and Information Technology,
Matunga, Mumbai, India
kaustubhchile@gmail.com

Vijay Sambhe
Veermata Jijabai Technological Institute,
Department of Computer Engineering and Information Technology,
Matunga, Mumbai, India
vksambhe@it.vjti.ac.in

*Corresponding author

Abstract *– Recent advances in computer vision have significantly impacted agriculture by enabling more precise monitoring of fruits. Deep learning techniques applied to fruit analysis enable the identification of fruit types and the detection of diseases, leading to improved yields and reduced environmental impact through timely intervention. However, most existing methods employ separate single-task models for each attribute (e.g., fruit classification or disease detection), resulting in increased complexity, higher computational cost, and the need for large labeled datasets for each task. In contrast, we propose a unified multi-task learning framework based on the Faster R-CNN architecture that simultaneously performs fruit type classification and disease detection within a single model. By sharing convolutional feature representations across both tasks, our approach leverages synergies between classification and detection, enhancing efficiency and accuracy. Training on a diverse dataset of fruit images annotated with both fruit identities and disease labels, our model jointly optimizes both tasks to learn more generalizable and discriminative features. Experimental evaluations demonstrate that this multi-task Faster R-CNN achieves competitive accuracy on both tasks while requiring fewer training samples and less computational resources than separate single-task models. The unified model not only simplifies system design but also accelerates inference and training, improving real-world scalability and robustness. This integrated approach provides a robust, efficient solution for automated fruit attribute analysis in agricultural applications.*

Keywords*: Multi-Task learning, Deep Learning, Convolutional Neural Networks, Attention Mechanism*

Received: September 23, 2025; Received in revised form: December 29, 2025; Accepted: December 29, 2025

1. INTRODUCTION

Mango, pomegranate, and guava are the most cultivated fruits globally with significant economic and nutritional importance. Mango is a vital fruit crop that contains essential vitamins and minerals and thrives in diverse agroecological regions worldwide. India observed an increase in the area used for mango orchards, and the production of mangoes in the past three decades, however a negative growth rate and higher instability was also observed [1] Mango yield is severely affected by insects that cause fruit and leaf disease [2]. However, pomegranate production has also increased in countries such as India, reaching 31,678 thousand tonnes in 2024, reflecting the growing demand for this fruit. In pomegranate, Pepitone is particularly popular

and contains abundant antioxidants and anti-inflammatory compounds. [3] Guava, widely grown in tropical and subtropical regions, with India being the largest producer, is another fruit of high economic and nutritional value. Typically round or oval, guava is rich in vitamin C, fiber, and antioxidants, which contribute to its popularity for fresh consumption and processing, as well as its notable health benefits. However, the prevalence of various diseases significantly limits mango production. Similarly, pomegranate fruit is highly susceptible to infection, leading to substantial losses. Guava cultivation faces several challenges similar to those encountered in mango and pomegranate production, particularly in terms of disease management. The main issues include canker, anthracnose, and root rot, all of which can severely impact yield and fruit quality.

Traditional methods of fruit quality assessment are labor-intensive and prone to human error. The subjective nature of visual inspections can lead to inconsistent results, making it challenging to implement standardized quality control measures. These limitations have led to the adoption of deep learning and computer vision techniques. Deep learning offers more accurate and efficient solutions for fruit and disease detection. By leveraging computer vision, real-time image analysis becomes feasible, enhancing operational efficiency in agriculture.

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), are widely used for image classification tasks in fruit and disease detection. Transfer learning with pre-trained models such as VGG16, ResNet, and InceptionV2 is commonly employed to enhance model performance. By applying object detection algorithms such as YOLOv3 and Faster R-CNN, it is possible to obtain precise fruit and diseased regions in images. Hybrid models that combine machine learning and deep learning techniques have shown high accuracy in fruit disease classification.

Despite advances in deep learning and computer vision in the field of agriculture, existing research often focuses on detecting a single fruit quality attributes such as fruit type, disease type, ripeness, size etc. To detect each feature, a separate model is trained independently, each using an extensive labeled dataset for each attribute, resulting in models performing single task and thereby increasing complexity and computational costs. Whereas for many agricultural activities such as fruit monitoring, harvesting, spraying, it is required to determine these features simultaneously. This is possible by training a single model to detect features like fruit type, disease type, ripeness, size etc. To address these limitations, multi-task learning models that leverage the shared features across individual tasks can be designed.

The main contributions of the present work are as follows: We propose a multi-task learning framework that integrates multiple tasks, such as fruit and disease detection and classification, implemented as a single model. This approach enhances efficiency by leveraging shared features across fruit type and disease detection tasks. It reduces the efforts of annotating individual datasets and also reduces training time for each task. By integrating multiple tasks, our framework provides a more comprehensive solution for assessing fruit quality. Also, we analyze the loss components of detection and classification process and apply dynamic weight updates needed for effective multitask learning. We have used feature sharing in an object detection model, with two classification and regression heads and dynamic update of weights of loss components during training enabling knowledge transfer between the complementary objectives

2. RELATED WORK

Remarkable success in implementing deep learning and computer vision has enabled extensive research on fruit and disease classification and detection over the past decade. Deep learning-based methods have achieved significant success in agriculture, by enabling precise fruit classification, disease detection, and yield prediction. In [2], a CNN network based on GoogleNet and VGG16 is trained on segmented images of mango leaves, using a softmax classifier to detect mango leaf diseases, achieving an accuracy of 98.67%. The results show that Mask R-CNN segmentation achieves better accuracy than K-means clustering. In [3], the authors proposed a hybrid CNN model optimized with the Honey Badger Optimization Algorithm (HBOA) and ResNet-50-based feature extraction, and integrated Detectron2 for diseased region segmentation to classify 700 pomegranate fruit images into four classes: anthracnose, bacterial blight, cercospora, and healthy. This model determines diseased regions using image segmentation with Mask R-CNN.

In this research [4], a multi-task learning (MTL) framework is developed to simultaneously perform fruit classification and freshness detection using a convolutional neural network (CNN). The model features a shared feature-extraction backbone (e.g., ResNet or VGG), followed by two separate branches: one for identifying the fruit type and another for assessing freshness. Each branch processes the extracted feature maps independently—fruit classification uses a fully connected layer with softmax activation, while freshness detection predicts quality levels through either a softmax classifier or a regression-based activation function. By sharing representations, this framework improves learning efficiency, enhances generalization, and achieves higher accuracy than single-task models.

In [5], the authors proposed a convolutional neural network (CNN) model to automatically detect citrus fruit and leaf diseases, including black spot, canker, scab, greening, and Melanose. Evaluated on the Citrus and PlantVillage datasets, the model achieved a test accuracy of 94.55%, outperforming state-of-the-art methods and providing a reliable tool for farmers. In [6], the research employs a transfer learning-based CNN architecture,

where a pre-trained model (e.g., ResNet, VGG, or Inception) extracts features, followed by fine-tuning layers for fruit classification and quality grading. The methodology involves image preprocessing (resizing, augmentation) and multi-task learning, with a shared feature extractor and task-specific output layers. The model is trained on a multi-fruit dataset containing labeled images for both classification and grading, utilizing the Adam optimizer and cross-entropy loss function. The proposed approach significantly enhances classification performance while reducing the need for extensive labeled datasets. A Faster R-CNN model with classifier fusion is proposed for the automatic detection of small fruits in [7], enhancing recognition performance by integrating multiple classifiers. By optimizing feature extraction through multi-scale learning, the model effectively addresses challenges posed by small fruit sizes and complex backgrounds. Experimental results demonstrate superior detection accuracy, highlighting the effectiveness of classifier fusion in improving recognition robustness.

In [8], the authors compare a YOLOv8 and CenterNet deep learning model for ripeness detection, The YOLO model outperforms in speed and accuracy Detection is performed on apples in room conditions. [9] replaces the backbone network with a moving-window transformer and introduces a new smoothing loss It achieves an excellent result on an apple dataset with a mAP 0.692. In [10], the authors proposed apple detection along with size estimation using a multitask network with Mask R-CNN and FPN. Feature maps from the FPN and depth data are fed into a regression head to estimate apple size. An F1-score of 0.88 and a mean absolute error of 5.64 mm in diameter estimation across over 15,000 annotated apples. While the method offers state-of-the-art accuracy and efficient inference using affordable hardware, it requires extensive annotated data and performs best on apples with more than 65% visibility. DenseNet network was used in [11] to assess the quality of six fruits, achieving an accuracy of 99.67%. A custom lightweight object detection model based on YOLO to detect melon ripeness is proposed in [12], capable of providing inference on edge devices with 85.9% mAP. In [13], the authors compared a transformer and an MLP for fruit ripeness detection. Swin Transformer achieved a precision of 87.43%. A pruned and quantized Faster-RCNN—fine-tuned via transfer learning and enhanced with RGB+NIR context data fusion—was employed in

Research efforts focused on lightweight models for disease detection have also yielded promising results. [14] on the Citrus fruit dataset for edge-based disease classification. The PFDI model achieved 96.92% accuracy unpruned, 96.03% after 60–90% pruning (28.16 MB), and 87.2% after pruning + 8-bit quantization (8.23 MB), making it ideal for deployment on memory-constrained edge devices. In [15], two models—a custom CNN and a modified MobileNetV2—were trained on a dataset comprising healthy and diseased cashew fruit and nuts, achieving test accuracies of 99.48% and over 98%, respectively. Similarly, [16] proposed an ultra-lightweight CNN with a three-level feature-reuse network containing just 101K parameters for plant disease detection on banana, guava, and mango leaf images, achieving an accuracy of 99.14% and surpassing 15 heavyweight pre-trained benchmarks. Leaf disease classification for apple and custard apple using AlexNet and SqueezeNet is presented in [17], where color images outperformed black and white and gray images. In [18], an ensemble model combining ResNet-50 and ResNet-101 backbones was proposed for multi-task fresh vs. rotten and fruit-type classification, achieving fruit classification accuracy of 99.78% and freshness classification accuracy of 99.06%. The features from the backbone networks are concatenated into a joint fully connected head and fed to a multi-task classifier.

Transformer-based architectures have recently gained traction in agricultural detection and classification tasks. For example, the PDLC-ViT model [19] employs a Vision Transformer-based multi-task learning framework for plant disease localization and classification, achieving 99.97% accuracy and a mean average precision of 99.18%, demonstrating excellent performance. In [20], a comparison between DETR (a transformer-based model) and YOLOv8 for citrus fruit detection showed YOLOv8 outperforming DETR in precision and recall, making it more effective for real-world applications. In most of these research papers, either a single task model has been used to perform only one task, or a multi-task learning approach has been adopted, but only to perform classification-related tasks and not detection tasks. Our model proposes a unified architecture for the classification and detection of fruit types and diseased regions. [21] discuss the dynamic loss optimization concept. [22, 23] improved disease and soil classification by applying modified state-of-the-art models and balancing techniques, emphasizing adaptation to specific agricultural contexts.

RESEARCH GAP

Most existing fruit analysis methods rely on separate single-task models for leaf or fruit disease classification [2, 3, 5, 6, 11, 17]. A multitask CNN using shared features for classification of freshness and fruit type is proposed in [4]. In [10], a similar CNN network is proposed for fruit and size estimation. A transformer based approach to classify and detect leaf diseases is developed by using shared features and attention mechanism. An ensemble network with two feature extractors are used in [18]. Our approach is motivated by [4] and [19], Lightweight object detection models have also been proposed for disease detection.[14, 15, 16] However our model is a detection model, a Faster RCNN network that detects fruit disease and fruit type with localization and labels. by leveraging shared feature representations across tasks and incorporating advanced modules—such as spatial/channel attention and multi-scale feature fusion—to enhance robustness. Recent studies show that integrating attention mechanisms into multitask networks can sig-

nificantly boost disease classification accuracy, and that combining attention with multi-scale fusion achieves high accuracy while maintaining real-time processing speeds. Despite these advances, a unified framework that efficiently handles both fruit classification and disease localization with improved accuracy and real-time performance is still missing. This gap motivates our proposed multitask Faster R-CNN framework, which is described in detail in the following Methodology section.

3. MATERIALS AND METHODS

Three Fruit disease datasets were considered for this research, all taken from Kaggle. Table I shows the various fruits and their corresponding diseases and Table II shows the distribution of images w.r.t fruit and disease. A sample of images was taken for both the train and test datasets. The training dataset comprises 263 images, and the test dataset comprises 65 images. The images were resized to a uniform dimension of 255 × 255, The images present in the train and test datasets were annotated using the VGG annotator. Two bounding boxes are defined for each image - one bounding box for the fruit depicting the fruit type, and the other bounding box for the diseased portion of the fruit depicting the disease type. Fig. 1 shows a few sample images from the dataset and their corresponding annotations.

Table 1. Fruits and their Diseases

Fruit	Disease
Mango	Stem and Rot, Healthy, Black Mould Rot, Alternaria
Pomegranate	Healthy, Cercospora, Bacterial Blight, Anthracnose, Alternaria
Guava	Scab, Phytophthora, Stem and Rot

4. MULTITASK FASTER RCNN ARCHITECTURE

Most existing fruit analysis methods use separate single-task models for classification or disease detection, rather than a joint approach. The MultiTaskFasterRCNN architecture, as shown in Fig. 2. is a specialized extension of the Faster R-CNN model [24] with a ResNet-50 backbone and Feature Pyramid Network (FPN), specifically tailored for the simultaneous detection and classification of fruits and their diseases.

4.1. MODEL OVERVIEW

The model handles three fruit classes (mango, guava, pomegranate) and seven disease classes (Alternaria, Black Mould rot, Healthy, Stem and Rot, Anthracnose, Bacterial Blight, Cercospora, Scab, Phytophthora). It employs a unified detection head with a custom box predictor that outputs class-specific bounding boxes and confidence scores for 12 classes (including background). The architecture incorporates several advanced components to enhance performance: attention mechanisms to highlight relevant features, multi-scale feature fusion to capture objects at different scales, and disease-specific enhancements to identify subtle disease patterns better. The model leverages transfer learning by initializing with pre-trained weights from a standard Faster R-CNN model and then fine-tuning it for the specific task of fruit disease detection. This approach combines the robust feature extraction capabilities of ResNet-50 with custom modules designed to address the challenges inherent to agricultural disease detection, including subtle visual symptoms, variations in disease appearance, and the need to distinguish between healthy and infected regions.

4.2. ATTENTION MECHANISM

4.2.1. Channel Attention

Channel attention is a critical enhancement to the base Faster R-CNN architecture that enables the model to focus on the most discriminative feature channels for fruit disease detection. In agricultural image analysis, various diseases exhibit unique color patterns, textures, and visual characteristics that are encoded in specific convolutional feature channels. The channel attention module explicitly models interdependencies between channels, amplifying important features while suppressing less useful ones.

The implementation follows a squeeze-and-excitation approach, where global spatial information is first "squeezed" into channel descriptors via global average pooling, creating channel-wise statistics that capture the global distribution. These global features are then passed through a dimensionality-reduction layer (using a 1 × 1 convolution to reduce from 256 to 64 channels), followed by a ReLU activation to introduce nonlinearity. After this, the features are expanded back to the original dimension through another 1×1 convolution (from 64 to 256 channels), with a sigmoid activation that produces channel attention weights in the range [0,1]. Mathematically, for an input feature map of width W, height H and channels C, $F \in R^{CxHxW}$, the channel attention process can be expressed as:

$$z = \frac{1}{HxW}\sum_{i=1}^{H}\sum_{j=1}^{W} F(:i,j) \quad (1)$$

$$z' = W_1(z) \quad (2)$$

$$z'' = \partial(z') \quad (3)$$

$$s = \sigma(W_2(z'') \quad (4)$$

$$F_{channel} = F \otimes s \quad (5)$$

Where $z \in RC \times 1 \times 1$ represents the global average pooled features, $W1 \in R\ 64 \times 256$ and $W2 \in R\ 256 \times 64$ are the learnable parameters of the dimension reduction and expansion convolutions, δ is the ReLU activation, σ is the sigmoid function, and $\otimes$ denotes channel-wise multiplication where each channel is scaled by its corresponding attention weight.

The channel attention mechanism is particularly valuable for fruit disease detection because different diseases affect visual characteristics in distinct ways.

Table 2. Distribution of images Fruit vs Disease (Train+Test)

Fruit	Disease								Healthy
	Stem and Rot	Black Mould Rot	Alternaria	Cercospora	Bacterial Blight	Anthracnose	Scab	Phytophthora	
Mango	26+06	25 + 05	20 + 04	---	---	---	---	---	22 + 05
Pomegranate	---	---	24+06	24+06	24+06	24+06	---	---	24+06
Guava	16 + 05	---	---	---	---	---	16 + 05	22 + 05	---

Fig.1. Annotated images of diseased mango,pomegranate and guava. For each image we have 2 bounding boxes- one for the fruit type and other for the disease type

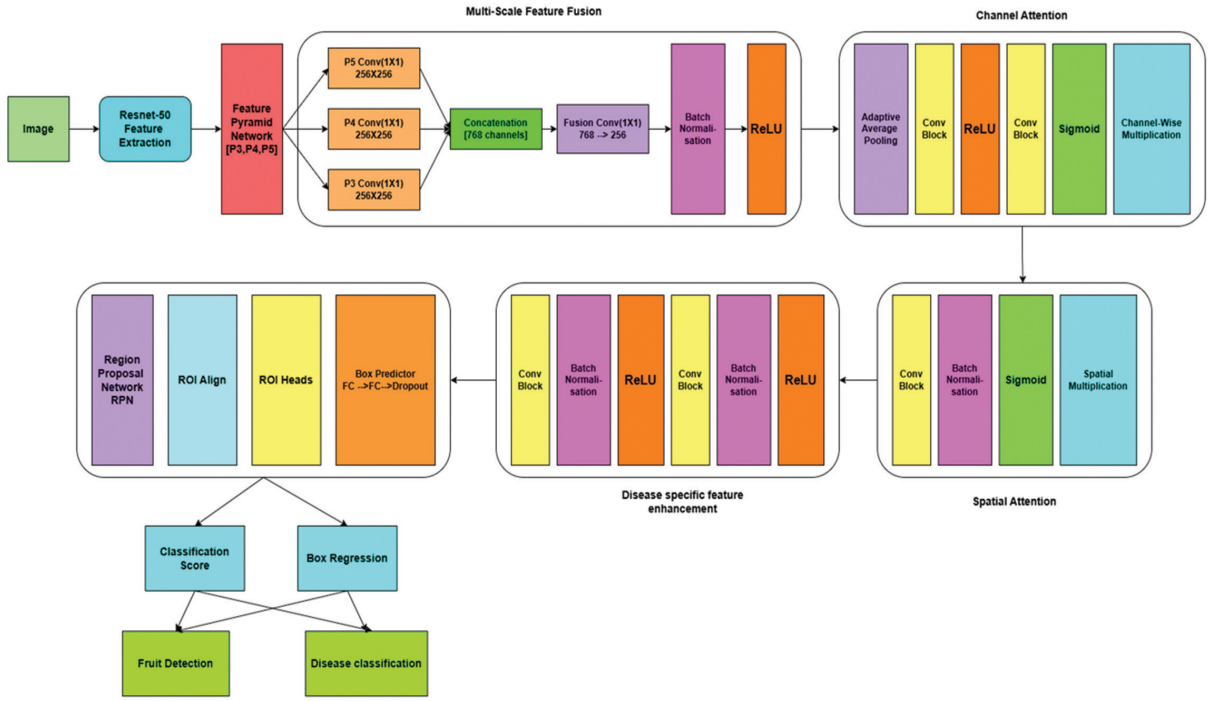

Fig. 2. Model Architecture: Multi-Task Faster RCNN for simultaneous classification and detection of Fruit and Diseased part

For example, some may cause color changes captured in specific channels, while others may manifest as textural changes represented in different channels. By learning to emphasize the most discriminative channels, the model becomes more sensitive to subtle disease-specific features that might otherwise be overlooked in the standard Faster R-CNN pipeline.

4.2.2. Spatial Attention

Spatial attention complements channel attention by focusing on "where" to look within the feature maps, highlighting spatially informative regions while suppressing less relevant areas. This mechanism is fundamental for fruit disease detection, as diseases often af-

fect only specific regions of the fruit (e.g., stem-end rot primarily occurs near the stem, while other diseases may present as scattered spots across the fruit surface). Unlike channel attention, which operates along the channel dimension, spatial attention operates across spatial dimensions to generate a 2D attention map that identifies the most informative regions in the feature representation. The implementation begins by aggregating channel information using a 7×7 convolutional layer that reduces the channel dimension to 1, effectively creating a spatial attention map. The large kernel size (7×7) allows the module to consider a broader spatial context when determining region importance. The spatial attention map is batch-normalized to stabilize training and normalize feature distributions, and followed by a sigmoid activation that produces spatial weights in the range [0,1]. For an input feature map F channel $\in R^{C\times H\times W}$ (already processed by channel attention), the spatial attention mechanism can be formulated as:

$$M = \sigma(BN(Conv_{7x7}F_{Channel})))$$
$$F_{spatial} = F_{channel} \otimes \mathrm{M}$$

Where $M \in R^{1\times H\times W}$ is the spatial attention mask, Conv7×7 denotes a 7×7 convolution with output channel dimension 1, BN is batch normalization, σ is the sigmoid function, and $\otimes$ represents element-wise multiplication where each spatial position in the feature map is weighted by its corresponding attention value. In the context of fruit disease detection, spatial attention helps the model focus on disease-affected regions that exhibit visual anomalies compared to healthy tissue. For instance, when examining fruits with localized infections or blemishes, spatial attention can highlight affected areas while suppressing background or healthy regions, thereby improving the model's ability to detect and classify diseases accurately. This attention mechanism is especially valuable when diseases present with subtle symptoms that might be overlooked by the model when considering the entire fruit, or when the disease affects only a small portion of the visible surface. The combination of channel and spatial attention creates a comprehensive attention mechanism that selectively emphasizes both"what" (relevant feature types) and"where" (informative regions) in the feature maps, significantly enhancing the model's disease detection capabilities.

4.2.3. Multi-scale Feature Fusion

The multi-scale feature fusion module integrates feature maps from multiple levels of the Feature Pyramid Network to capture and leverage information across different spatial resolutions. In fruit disease detection, this is crucial because diseases can appear at at different scales, from small, localized spots to large damaged areas, and depending on the fruit variety and camera distance, fruits can appear at various sizes in images.

Feature maps P3, P4, and P5, which represent features at high, medium, and low resolutions, respectively, are first extracted from the backbone network. P4 and P5 are upsampled using bilinear interpolation to match the spatial dimensions of P3 (the highest resolution) in order to facilitate direct fusion. Before fusion, each feature map is subjected to a level-specific 1×1 convolution to normalize feature representations.

This fusion approach improves the model's capability to detect disease patterns, regardless of their scale. By combining complementary information from different resolution levels within the feature pyramid, it increases the model's robustness to variations in imaging conditions and disease manifestations.

Mathematically, the multi-scale feature fusion can be represented as:

$$P_4' = Upsample(P_4, size(P_3)) \quad (6)$$

$$P_5' = Upsample(P_3, size(P_3)) \quad (7)$$

$$P_3^{(c)} = Conv_{1x1}^{(3)}(P_3) \quad (8)$$

$$P_4^{(c)} = Conv_{1x1}^{(4)}(P_4') \quad (9)$$

$$P_5^{(c)} = Conv_{1x1}^{(5)}(P_5') \quad (10)$$

$$F_{concat} = P_3^{(c)} \otimes P_4^{(c)} \otimes P_5^{(c)} \quad (11)$$

$$F_{fused} = \sigma(BN\left(Conv_{1x1}^{fusion}(F_{concat})\right)) \quad (12)$$

where:

$\oplus$ denotes channel-wise concatenation

$Conv^{(i)}_{1x1}$ represents the 1 × 1 convolution operation for level i

BN denotes batch normalization

σ represents the ReLU activation function:

$\sigma(x) = \max(0, x)$

size(P3) refers to the spatial dimensions of feature map P3

4.2.4. Disease-specific Feature Enhancement

The disease-specific feature enhancement module optimizes features to more effectively identify subtle disease patterns and symptoms, which is crucial for precise disease classification. Agricultural diseases frequently exhibit subtle visual indicators that are easily overlooked by conventional object detection networks, requiring the implementation of specialized feature enhancement techniques.

The enhancement process employs a sequence of two 3 × 3 convolutional layers with batch normalization and ReLU activations, creating a mini-residual network that deepens the feature representation:

$$F_1 = \sigma((BN_1(Conv_{3x3}^{(1)}(F_{input}))) \quad (13)$$

$$F_{enhanced} = \sigma((BN_2(Conv_{3x3}^{(2)}(F_1))) \quad (14)$$

where:

F_{input} is the input feature map from the attention mechanisms

$Conv^{(i)}_{3x3}$ represents the i-th 3×3 convolutional operation with padding=1

BN_i denotes the i-th batch normalization operation

σ represents the ReLU activation function:
$\sigma(x) = \max(0,x)$

Both convolutions maintain spatial dimensions:
256 x 256 channels

The mathematical formulation can also be expressed in functional form as:

$$F_{enhanced} = H_{disease}(F_{input}) \quad (15)$$

where $H_{disease}$ represents the complete disease enhancement transformation:

$$H_{disease}(x)=\sigma(BN_2(Conv^{(2)}_{3x3}(\sigma(BN1(Conv^{(1)}_{3x3}(x)))))) \quad (16)$$

This specialized enhancement enhances the model's ability to differentiate between visually similar diseases by highlighting disease-specific textural and color patterns through a sequential process of convolution, normalization, and nonlinear activation operations.

4.2.5. Forward Pass Integration

The forward pass integrates all specialized components into a cohesive workflow that enhances the base Faster R-CNN architecture. During training, the process begins with passing the input images through the backbone network (ResNet-50 with FPN) to extract hierarchical feature maps. These features are then passed through the multi-scale feature fusion module, which combines information from P3, P4, and P5 levels to create a unified feature representation that captures objects at different scales. The channel and spatial attention mechanisms, which highlight the most informative channels and spatial regions, are then applied sequentially to improve the fused features. The enhanced features are further refined by the disease-specific enhancement module, which emphasizes patterns relevant to disease identification. This enhanced feature map is being fed into the rest of the Faster RCNN pipeline, including the Region Proposal Network (RPN) and the detection heads. During inference, the model follows a similar path but returns detection results rather than computing losses. The forward method also includes comprehensive error handling to catch and report issues during the forward pass, particularly when dealing with potentially variable agricultural image data. The method returns the computed losses (classification, regression, RPN classification, and RPN regression) for training. For inference, it returns detection results, which include bounding boxes , class labels, and confidence scores.

4.2.6. Loss Function

The MultiTask FasterRCNN utilizes the standard Faster RCNN multi-task loss function, combining several components to train both region proposal and detection aspects simultaneously. The total loss is:

$$L_{total} = w_{rpn\_cls}L_{rpn\_cls} + w_{rpn\_reg}L_{rpn\_reg} + w_{roi\_cls}L_{roi_{cls}} + w_{roi\_reg}L_{roi\_reg} \quad (17)$$

Where

L_{rpn_cls} is the RPN classification loss

L_{rpn_reg} is the RPN regression loss

L_{roi_cls} is the ROI classification loss

L_{roi_reg} is the ROI regression loss

RPN is Region Proposal Network

ROI is Region of Interest

w_{rpn_cls}, w_{rpn_reg}, w_{roi_cls} and w_{roi_reg} are the weights of the respective RPN and ROI classification and regression losses

Each component addresses a specific aspect of the detection task:

RPN Classification Loss (Objectness Loss)

(L_{rpn_cls}): The RPN (Region Proposal Network) classification loss is a binary cross entropy loss that determines whether each anchor box contains an object or is just background. This is the first stage of object detection where the model learns to identify potential object locations.:

$$L_{rpn\_cls} = -\frac{1}{N_{cls}}\sum_{i=0}^{C} [y_i log(p_i) + (1 - y_i) log\,(1 - p_i)]$$

where y_i is the ground truth (1 for object, 0 for background) ,p_i is the predicted probability and N_{cls} is the number of classes.

RPN Regression Loss (L_{rpn_reg}) : Refines the initial anchor boxes to better align with actual object boundaries. This loss teaches the network to adjust anchor box coordinates to more precisely localize objects.

$$L_{rpn\_reg} = \frac{1}{N}\sum_i \quad smooth_{L1}(t_i - t_i^*)$$

where t_i represents predicted box coordinates and t_i* represents ground truth.

ROI Classification Loss (Region of Interest Classification Loss) (L_{roi_cls}): Multi-class cross-entropy for fruit and disease classification to classify detected objects into specific categories (mango, pomegranate, guava, and various diseases). This is the second stage where the model determines what type of object each region proposal contains. :

$$L_{roi\_cls} = -\frac{1}{N_{cls}}\sum_i \quad \sum_{c=1}^{C} [y_{ic} log\,(p_{ic})$$

where y_{ic} is 1 if instance i belongs to class *c* and 0 otherwise, and p_{ic} is the predicted probability.

ROI Regression Loss (Bounding box Regression Loss) (L_{roi_reg}): Further refines bounding boxes for each specific class after classification. This provides class-specific box adjustments, as different object types may require different refinement strategies.

$$L_{roi_{reg}} = -\frac{1}{N_{reg}} \sum_{i} \sum_{c=1}^{C} [y_{ic}.smooth_{L1}(t_i - t_i^*)]$$

where t_{ic} represents predicted box coordinates for class c and t^*_{ic} represents ground truth.

4.2.7. Dynamic Weighted Loss Function

Traditional multitask learning in object detection often suffers from an imbalance in losses, with some components dominating the training process and leading to suboptimal performance across tasks. [20] Modern object detectors (e.g., those built on Feature Pyramid Networks) perform predictions at multiple feature-map scales but treat each scale's loss equally during training. In practice, however, different scales exhibit uneven and fluctuating loss behaviors, leading to an objective imbalance where some scales dominate learning while others lag. In Faster R-CNN architectures, the standard loss function comprises four primary components: classification loss (L_{cls}), bounding box regression loss (L_{bbox}), objectness loss (L_{obj}), and RPN box regression loss (L_{rpn}). However, these components typically exhibit vastly different magnitudes and convergence characteristics, with regression losses often dominating due to their continuous nature, while classification losses remain undertrained.

To address this fundamental challenge, we propose a dynamic loss weighting mechanism that adaptively adjusts loss component weights based on their relative contributions, convergence patterns, and temporal trends during training. Our approach operates on the principle of maintaining balanced optimization across all task objectives while preventing any single component from monopolizing the learning process. The system continuously monitors the loss ratio for each component i at epoch t:

$$r_i^{(t)} = \frac{L_i^{(t)}}{\sum_{j=1}^{N} L_j^{(t)}}$$

where $L^{(t)}_i$ represents the individual loss values and N=4 denotes the number of loss components. Additionally, we compute the loss trend over a stability window k to capture convergence behavior:

$$\Delta L_i^{(t)} = \frac{1}{k} \sum_{m=0}^{k-1} L_i^{(t-m)} - \frac{1}{k} \sum_{m=1}^{k} L_i^{(t-m)}$$

This formulation uses overlapping windows to compute the trend, where the recent window spans epochs (t–k+1) to t, and the comparison window spans epochs ($t-k$) to ($t-1$), providing a more stable trend estimation with smaller time lags. The dynamic weight update mechanism is governed by a piecewise function that responds to both magnitude imbalance and convergence status:

$$w_i^{(t+1)} = \begin{cases} w_i^{(t)}.(1+\alpha) \; if \; r^{(t)} > \tau_{high} \; and \; \Delta L_i^{(t)} \geq 0 \\ w_i^{(t)}.(1-\alpha.\gamma) \; if \; r^{(t)} < \tau_{low} \; and \; \Delta L_i^{(t)} < -\epsilon \\ w_i^{(t)} otherwise \end{cases}$$

where α = 0.1 is the primary adjustment factor for increasing weights of dominating components that show stagnation, γ = 0.5 is a damping coefficient to ensure gradual adjustments, τ_{high} = 0.3 and τ_{low} = 0.15 represent the ratio thresholds for triggering weight modifications, and ϵ = 0.001 defines the convergence threshold for trend analysis.

Furthermore, to prevent total loss scale drift during training, we apply periodic normalization to the core detection components:

$$\hat{w}_i^{(t)} = w_i^{(t)}.\frac{4}{\sum_{j \in C} w_j^{(t)}}$$

The final weighted loss function applies the dynamic weights directly to the standard Faster R-CNN loss components:

$$L_{total}^{(t)} = \sum_{i \in C} \hat{w}_i^{(t)} L_i^{(t)}$$

where $C = \{classification, bbox\ regression, objectness, rpn\ bbox\ regression\}$ represents the core detection loss components.

The proposed mechanism operates with a warmup period of T_{warmup} = 2 epochs to allow initial loss stabilization before initiating adaptive adjustments, and employs a stability window of k = 3 epochs for reliable trend estimation while maintaining responsiveness to training dynamics.

5. EXPERIMENTS AND RESULTS

5.1. EXPERIMENTS

The goal of this research was to prove the superiority of multi-task learning over single-task approaches for agricultural disease detection. By simultaneously optimizing for fruit classification and disease identification within a unified Faster R-CNN framework, our model achieves improved performance on both tasks. We compared our multitask framework for fruit and disease detection and classification with a single-task framework for detecting and classifying fruit and disease separately. We computed the accuracy, precision, recall, and F1 score for each scenario for comparative analysis.

Our research leveraged multiple pretrained backbone architectures integrated with Feature Pyramid Networks (FPN) to identify the optimal foundation for our multi-task learning approach. While ResNet-50 with FPN served as our primary architecture due to its balanced performance-efficiency trade-off, we conducted a comprehensive evaluation across several popular backbone networks to empirically validate our design choices. Each backbone variant was integrated with our custom multi-task detection framework while maintaining consistent parameter settings across experiments. Performance was rigorously evaluated using accuracy, precision, recall, and F1 score—calculated separately for both fruit classification and disease detection. This compara-

tive analysis as shown in Table 3 offered explanations for how different feature-extraction architectures affect the model's ability to perform both tasks simultaneously.

5.2. PARAMETER SETTINGS

In our implementation of the multi-task Faster R-CNN model for fruit disease detection, we configured key hyperparameters based on empirical testing and domain-specific requirements. We selected the AdamW optimizer with a learning rate of 0.0001 and weight decay of 0.0005 to balance stable convergence with effective regularization for our complex dual-task architecture. The relatively small batch size of four was chosen to accommodate GPU memory constraints while still allowing frequent weight updates, which is beneficial for learning the nuanced features of diseased fruit. Our backbone architecture uses a pre-trained ResNet-50 with FPN to leverage transfer learning and address multi-scale detection challenges. The custom attention mechanisms use a computationally efficient bottleneck design (256→64→256) for channel attention and a 7×7 spatial attention kernel to capture appropriate contextual information for subtle disease symptoms. During evaluation, we applied a confidence threshold of 0.5 for balanced metrics reporting, while lowering the threshold to 0.3 during inference to prioritize recall in agricultural applications where missing diseased fruits could have significant consequences. This parameter configuration optimizes the model's ability to simultaneously perform fruit identification and disease detection while maintaining computational efficiency.

5.3. EVALUATION RESULTS

Our multitasking Faster R-CNN model processes fruit images through a comprehensive pipeline that simultaneously detects fruits and identifies diseases. When an input image enters the system, it first passes through a ResNet-50 backbone with Feature Pyramid Network, which extracts hierarchical feature representations at multiple scales. These features are then enhanced by our custom-designed attention mechanisms—channel attention highlights relevant feature channels, while spatial attention emphasizes important regions within the feature maps.

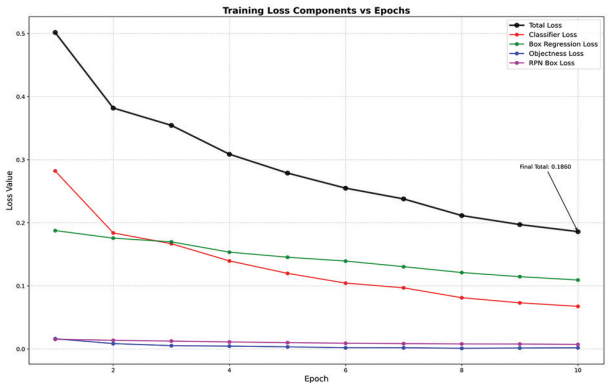

Fig. 3. Training Loss Components vs Epochs

During inference, the model outputs bounding boxes with associated class probabilities for both fruit categories and disease conditions. The system applies confidence thresholds to filter predictions: fruit detection uses 0.3 to maximize recall, while disease classification uses 0.5 to ensure reliability in agricultural applications.

The performance metrics Accuracy, Precision, Recall and F1 score are used to evaluate classification results.

$$Precision = TP/(TP + FP)$$

$$Recall = TP/(TP + FN)$$

$$F1\ Score = \frac{2x(PrecisionxRecall)}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where

TP = True Positives

TN =True Negatives

FP= False Positives

FN=False Negatives

The Epoch-level Training Loss as shown in Fig. 3. graph displays the overall training loss and it's components measured at the epoch level across 10 complete training epochs. It demonstrates the model's learning curve and convergence pattern at a broader scale, showing how the loss steadily decreases as training progresses toward the minimum value of 0.1725.

The graph Fig .4 shows the evolution of dynamic loss weights during multi-task object detection training, where the system automatically rebalances loss components based on their convergence rates and relative magnitudes The RPN bbox regression weight increases from 1.0 to 1.06 while the bbox regression weight decreases to 0.91, demonstrating the adaptive weighting strategy that emphasizes region proposal network training in later epochs while reducing focus on box regression as spatial localization improves.

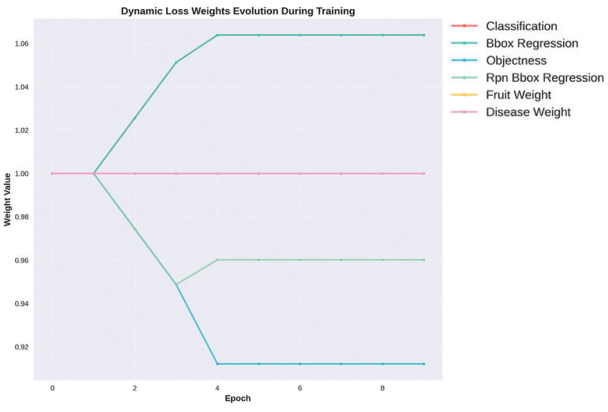

Fig. 4. Dynamic loss weights automatically rebalance during multi-task object detection training

The Component-wise Training Loss over batches as shown in Fig. 5 shows how different loss components (total loss, classifier loss, box regression loss, objectness loss, and RPN box loss) decrease over training batches. It illustrates which components contribute most significantly to the overall loss and how quickly each component converges during training.

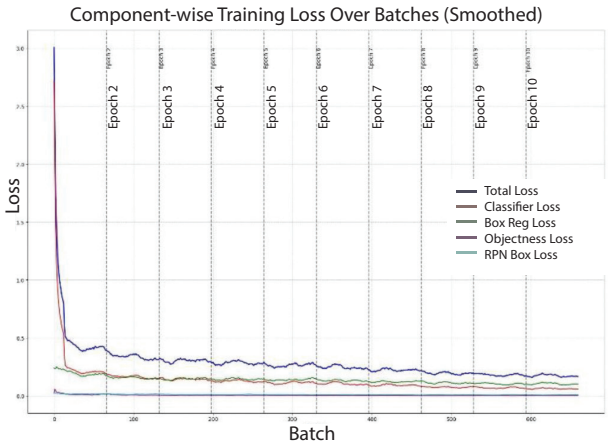

Fig. 5. Component-wise Training Loss Over Batches

The predicted and actual labels and bounding boxes for three images are shown in Fig. 6. The model achieves 95.38% accuracy, 95.99% precision, and 95.91% F1 score, demonstrating its fruit classification capabilities. For the more challenging task of disease detection, the model achieves strong performance, with an accuracy and recall of 83.08%, an F1 score of 83.51%, and a Precision of 89.21%. The mAP is calculated by first computing the Average Precision (AP) for each class at multiple IoU thresholds (0.5-0.95), where AP measures the area under the precision-recall curve. These AP values are then averaged across all IoU thresholds for each class, and finally averaged across all classes to obtain the mean Average Precision.

Table 2. Performance Metrics Comparison Between SingleTask and Multi-Task Frameworks

Metric	ST Fruit	ST Disease	MT Fruit	MT Disease
Accuracy	93.85%	66.15%	93.85%	83.08%
Precision	95.14%	61.74%	94.50%	85.21%
Recall	93.85%	66.15%	93.85%	83.08%
F1 Score	93.99%	60.34%	93.55%	82.74%

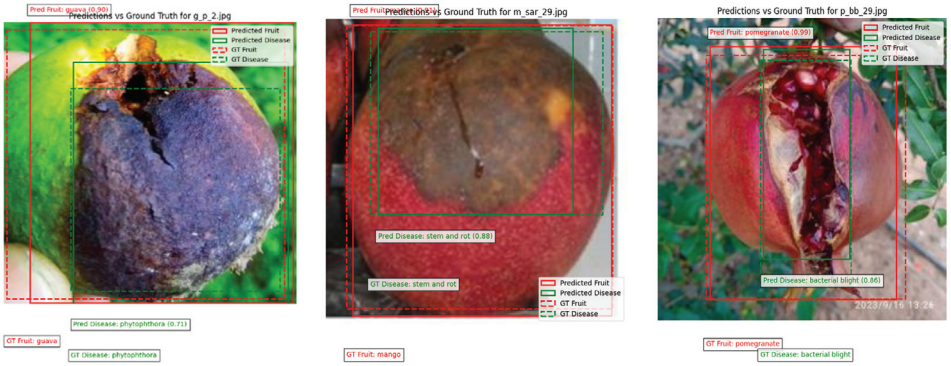

Fig. 6. Showing the predicted and actual bounding boxes and labels for each fruit

The evaluation results demonstrate strong performance in fruit detection, with a **mAP of 0.752**, indicating robust localization and classification. In contrast, disease detection achieved a lower mAP of 0.39. These metrics validate the effectiveness of our multitask learning approach.

Based on the F1 score curves shown in Fig. 7, our multitask model demonstrates exceptional performance across both fruit classification and disease detection tasks. For fruit classification, the model achieves perfect F1 scores (1.000) for all three fruit types—mango, pomegranate, and guava—with the micro-average F1 score reaching 0.985. The curves show that optimal performance is maintained across a relatively wide threshold range (approximately 0.2-0.4), indicating robust and reliable fruit detection. The disease classification results in Fig. 8 are equally impressive, with several disease classes achieving perfect F1 scores of 1.000 including black mould rot, healthy samples, bacterial blight, scab, and phytophthora.

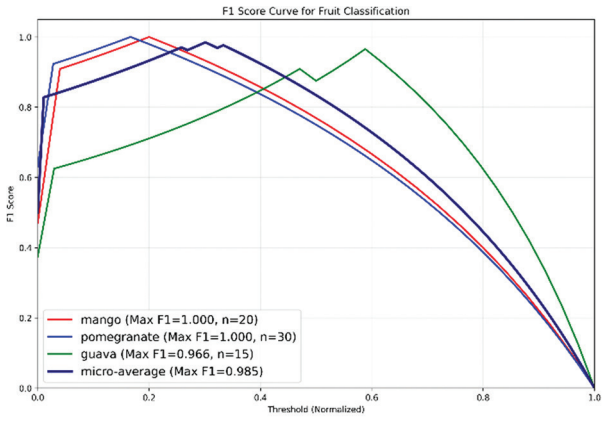

Fig.7. F1 Score vs. Confidence Threshold for Fruit Classification

Even the more challenging disease classes, such as alternaria, stem and rot, anthracnose, and cercospora, demonstrate strong performance, with F1 scores ranging from 0.833 to 0.909. The micro-average F1 score of

0.887 for disease conditions classification reflects different model's despite the inherent complexity and visual similarity among specific disease symptoms. Fig. 9 and Fig. 10 show the confusion matrix for each task respectively.

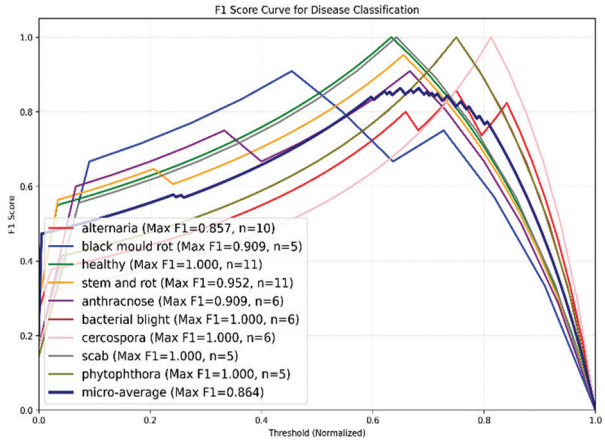

Fig. 8. F1 Score vs. Confidence Threshold for Disease Classification

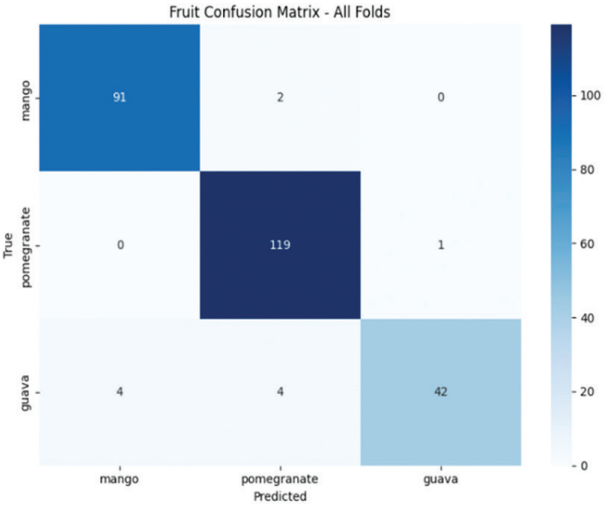

Fig. 9. Fruit Confusion Matrix

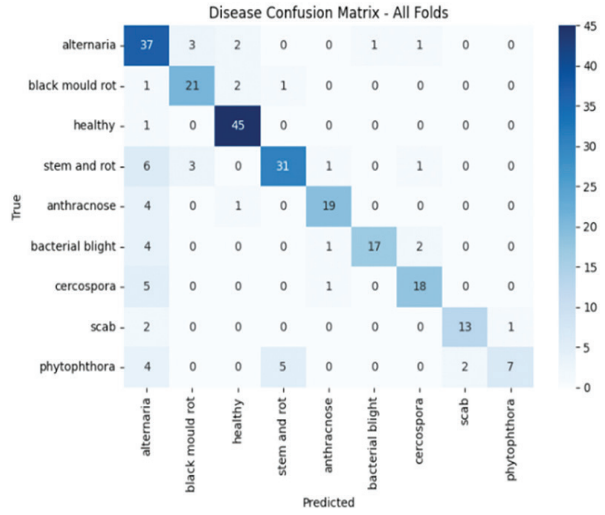

Fig. 10. Disease Confusion Matrix

5.4. COMPARISON WITH SINGLE-TASK FRAMEWORK

To determine whether the multi-task framework using Faster R-CNN improves the accuracy of fruit and disease detection and classification, we created single-task frameworks with architectures similar to our multi-task framework. These models contained a pretrained Faster R-CNN with ResNet-50 and FPN, along with an attention mechanism, and handled fruit and disease detection as independent tasks. The Accuracy, Precision, Recall, and F1 score for single-task and multi-task detection and classification of Fruit and Disease are shown in Table 2.

The single-task models were trained separately on the same dataset, with one model dedicated to fruit detection and another to disease identification.

Comparative analysis revealed that our multitask learning approach not only reduced computational overhead by combining the tasks into a unified architecture but also achieved superior performance metrics, particularly in disease classification, where contextual information from fruit detection proved beneficial.

5.5. COMPARISON WITH MULTIPLE CNN BACKBONES

EfficientNet-B0, ConvNeXt-Tiny, Inception-V3, and ResNeXt-50 (32×4d). The comparisons of the various backbones integrated with FPN are shown in Table 3.

Expanding, each backbone was carefully integrated with FPN to maintain consistent multi-scale feature extraction capabilities while varying the underlying feature extraction architecture. The results revealed significant performance variations across backbones, with Resnet50 surprisingly outperforming all other architectures in disease detection with 84.62% accuracy while maintaining an accuracy of 98.46% for fruit classification.

In contrast, Inception--v3 and EfficientNet-B0 achieved substantially lower disease detection performance, with accuracies of 52\% and 34\%, respectively. MobileNet-v3 demonstrated a strong balance of speed and accuracy for both fruit and disease classification. The EfficientNet-B0 also achieved a very poor accuracy of 35.38% for fruit classification.

5.6. K-FOLD CROSS VALIDATION

The performance of the proposed model was evaluated using three validation strategies: a single train–test split (without K-fold), 5-fold cross-validation, and 10-fold cross-validation. The results demonstrate clear differences in estimated performance depending on the evaluation protocol. The single-split evaluation produced the highest scores across all metrics, with Precision, Recall, Accuracy, and F1-Score all exceeding 0.984. However, these values are likely optimistic because a single split does not adequately capture variability in

data distribution and may therefore overestimate the model's generalization capability. Table 4 and Table 5 show the k-fold results.

In comparison, the 5-fold cross-validation results exhibit a moderate decline in all performance metrics (e.g., Precision = 0.9454, Recall = 0.9198, Accuracy = 0.9198, F1-Score = 0.9124). Notably, the variance values for 5-fold cross-validation remain extremely low (0.00024–0.00037), indicating that the model performs consistently across folds and is not highly sensitive to specific data partitions.

The 10-fold cross-validation results reveal improved performance relative to the 5-fold evaluation, with Precision, Recall, Accuracy, and F1-Score increasing to 0.9814, 0.9732, 0.9732, and 0.9682, respectively. This improvement can be attributed to the larger proportion of training data available in each fold, enabling the model to learn more representative patterns. Although the variance values for 10-fold cross-validation are slightly higher than those of the 5-fold setting (0.00114–0.00144), they remain minimal, suggesting strong stability and robustness across different data partitions.

Overall, the cross-validation results demonstrate that while single-split evaluation produces inflated performance estimates, both 5-fold and 10-fold cross-validation offer more reliable and generalizable assessments. Among these, 10-fold cross-validation provides the most balanced trade-off, yielding high performance with consistently low variance, thereby serving as the most dependable indicator of true model generalization.

Table 3. Performance Comparison of Different Backbones

Backbone	Fruit				Disease			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
ResNet-50+FPN	0.9846	0.9856	0.9846	0.9847	0.8462	0.8918	0.8462	0.8397
MobileNetV3+FPN	0.9846	0.9853	0.9846	0.9845	0.8	0.8615	0.8	0.7958
ResNet-101+FPN	0.9846	0.9853	0.9846	0.9845	0.5231	0.5074	0.5231	0.4563
EfficientNet-B0+FPN	0.3538	0.5608	0.3538	0.234	0.3385	0.3404	0.3385	0.2496
ConvNeXt-Tiny+FPN	0.8462	0.8783	0.8462	0.8168	0.6923	0.6602	0.6923	0.6502
Inception-V3+FPN	0.8462	0.8974	0.8462	0.8386	0.2308	0.2037	0.2308	0.1475
ResNeXt-50 (32×4d)+FPN	0.6308	0.7423	0.6308	0.6445	0.3385	0.2627	0.3385	0.2628

Table 4. k-fold validation results for Fruit detection

	Without K-fold	5 Folds	Variance 5 folds	10 Folds	Variance 10 folds
Precision	0.9853	0.9454	0.00024	0.9814	0.00114
Recall	0.9846	0.9198	0.00037	0.9732	0.00144
Accuracy	0.9846	0.9198	0.00037	0.9732	0.00144
F1 Score	0.9845	0.9124	0.00036	0.9682	0.00129

Table 5. k-fold validation results for Disease detection

	Without K-fold	5 Folds	Variance 5 folds	10 Folds	Variance 10 folds
Precision	0.745	0.8317	0.00293	0.8317	0.00364
Recall	0.7692	0.7719	0.00222	0.8204	0.00601
Accuracy	0.7692	0.7692	0.00222	0.8204	0.00601
F1 Score	0.7463	0.7677	0.00302	0.8081	0.00425

5.7. COMPARISON WITH MASK RCNN

When applying the Mask R-CNN model to the test dataset of 65 images, the Fruit Detection Accuracy is 1.000, and the Disease Detection Accuracy is 0.7000. In contrast, with the Faster R-CNN model, the Fruit Detection Accuracy is 1.000, and the Disease Detection Accuracy is 0.9000. F1 score for Mask R-CNN is 0.9355 for fruit detection, and for Faster R-CNN, F1 score for Fruit Detection is 0.9845, and for Disease Detection, Mask RCNN F1 score is 0.7067 and Faster RCNN F1 Score is 0.7943. This shows that the Faster R-CNN model provides better results than Mask R-CNN.

5.8. VISUALIZATION

The class activation maps obtained by applying the Grad-CAM method to the feature maps of the three fruits are shown in Fig. 11. The highlights in the diseased regions indicate that the model extracts features relevant to diseases and fruits.

6. CONCLUSION

In this research, we introduced a unified multi-task learning framework based on Faster R-CNN for simultaneous fruit classification and disease detection, addressing key limitations of traditional single-task models. By leveraging shared feature representations via a ResNet-50 + FPN backbone, multi-scale feature fusion, and custom attention mechanisms, our model effectively reduces computational overhead and minimizes the need for large, task-specific datasets. Experimental results demonstrated outstanding fruit classification performance and strong disease detection metrics, validating the benefits of multi-task learning in agricultural applications. Despite the inherent challenges of detecting subtle disease patterns, our approach achieved high recall rates, a crucial factor for real-world deployment. Overall, the proposed framework offers a robust, efficient, and scalable solution for automated fruit quality assessment, paving the way for more intelligent, real-

time agricultural monitoring systems. The analysis of loss components and, dynamic weight updates are important for the performance of multitask learning

Future work can increase the dataset size and include more number of images for a wider range of fruits and diseases. Collecting and annotating disease data for a specific fruit is crucial since the model is trained on both,fruit and disease type simultaneously.

Also optimizing lightweight architectures for edge device deployment is necessary .

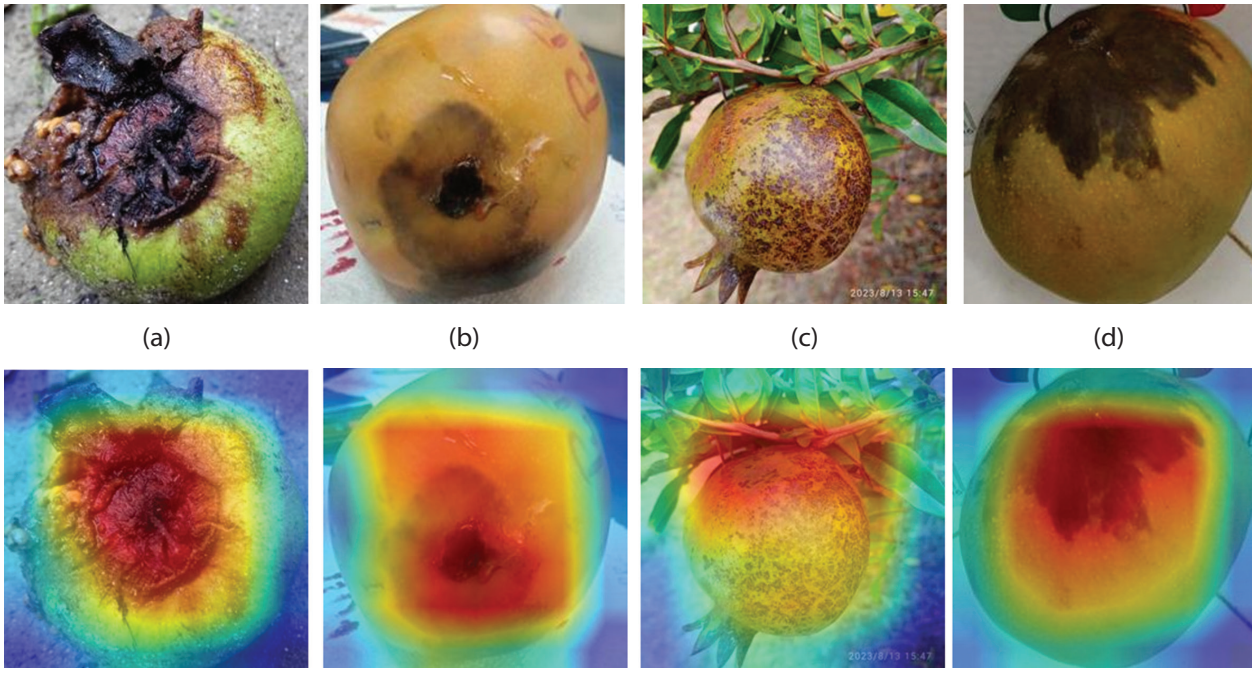

Fig.11. Visualization of Detection Results applied on Guava, Mango and Pomegranate. First row images are (**a**) Guava-Phytophthora (**b**) mango-black mould rot (**c**) Pomegranate-Cercospora (**d**) Mango-Stem ad Rot. Second row are the heatmap images for the respective images

REFERENCES

[1] G. Balaganesh, G. Makarabbi, "An Analysis on Performance of Mango Production in India", Asian Journal of Agricultural Extension, Economics & Sociology, Vol. 41, No. 10, 2023, pp. 968-976.

[2] Y. A. Bezabh, A. M. Ayalew, B. M. Abuhayi, T. N. Demlie, E. A. Awoke, T. E. Mengistu, "Classification of Mango Disease Using Ensemble Convolutional Neural Network", Smart Agricultural Technology, Vol. 8, 2024, p. 100476.

[3] P. Sameera, A. A. Deshpande, "Disease Detection and Classification in Pomegranate Fruit Using Hybrid Convolutional Neural Network with Honey Badger Optimization Algorithm", International Journal of Food Properties, Vol. 27, No. 1, 2024, pp. 815-837.

[4] Y. Zhang, X. Yang, Y. Cheng, X. Wu, X. Sun, R. Hou, H. Wang, "Fruit Freshness Detection Based on MultiTask Convolutional Neural Network", Current Research in Food Science, Vol. 8, 2024, p. 100733.

[5] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, A. Gumaei, "Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model", IEEE Access, Vol. 9, 2021, pp. 112942-112954.

[6] L. A. Aldakhil, A. A. Almutairi, "Multi-Fruit Classification and Grading Using a Same-Domain Transfer Learning Approach", IEEE Access, Vol. 12, 2024, pp. 44960-44971.

[7] X. Mai, H. Zhang, X. Jia, Max Q.-H. Meng, "Faster RCNN With Classifier Fusion for Automatic Detection of Small Fruits" ,IEEE Transactions on Automation Science and Engineering, Vol. 17, No. 3, July 2020 pp. 1555-1569.

[8] B. Xiao, M. Nguyen, W. Q. Yan, "Fruit Ripeness Identification Using YOLOv8 Model" Multimedia Tools and Applications, Vol. 83, No. 9, 2024, pp. 28039-28056.

[9] X. Kong, X. Li, X. Zhu, Z. Guo, L. Zeng, "Detection Model Based on Improved FasterRCNN in Apple

Orchard Environment", Intelligent Systems with Applications, Vol. 21, 2024, p. 200325.

[10] M. Ferrer-Ferrer, J. Ruiz-Hidalgo, E. Gregorio, V. Vilaplana, J. R. Morros, J. Gene-Mola, "Simultaneous Fruit Detection and Size Estimation Using Multitask Deep Neural Networks", Biosystems Engineering, Vol. 233, 2023, pp. 63-75.

[11] Md. S. Morshed, S. Ahmed, T. Ahmed, M. U. Islam, A. B. M. A. Rahman, "Fruit Quality Assessment with Densely Connected Convolutional Neural Network", Proceedings of the 12th International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, 21-23 December, 2022.

[12] X. Jing, Y. Wang, D. Li, W. Pan, "Melon Ripeness Detection by an Improved Object Detection Algorithm for Resource Constrained Environments", Plant Methods, Vol. 20, No. 1, 2024, p.127.

[13] B. Xiao, M. Nguyen, W. Q. Yan, "Fruit Ripeness Identification Using Transformers", Applied Intelligence, Vol. 53, 2023, pp. 22488-22499.

[14] P. Dhiman, P. Manoharan, U. K. Lilhore, R. Alroobaea, A. Kaur, C. Iwendi, M. Alsafyani, A. M. Baqasah, K. Raahemifar, "PFDI: A Precise Fruit Disease Identification Model Based on Context Data Fusion with Faster-CNN in Edge Computing Environment", EURASIP Journal on Advances in Signal Processing, Vol. 72, 2023.

[15] K. G. Panchbhai, M. G. Lanjewar, V. V. Malik, P. Charanarur, "Small Size CNN (CAS-CNN), and Modified MobileNetV2 (CAS-MODMOBNET) to Identify Cashew Nut and Fruit Diseases", Multimedia Tools and Applications, Vol. 83, 2024, pp. 89871-89891

[16] Pragya Hari, Maheshwari Prasad Singh, "A Lightweight Convolutional Neural Network for Disease Detection of Fruit Leaves", Neural Computing and Applications, Vol. 35, No. 20, 2023, pp. 14855-14866.

[17] S. S. Gaikwad, S. S. Rumma, M. Hangarge, "Fungi Affected Fruit Leaf Disease Classification Using Deep CNN Architecture", International Journal of Information Technology, Vol. 14, 2022, pp. 3815-3824.

[18] J. Kang, J. Gwak, "Ensemble of Multi-Task Deep Convolutional Neural Networks Using Transfer Learning for Fruit Freshness Classification", Multimedia Tools and Applications, Vol. 81, 2021, pp. 22355-22377.

[19] S. Hemalatha, J. J. B. Jayachandran, "A Multitask Learning-Based Vision Transformer for Plant Disease Localization and Classification", International Journal of Computational Intelligence Systems, Vol. 17, 2024, p. 188.

[20] Z. Jrondi, A. Moussaid, M. Y. Hadi, "Exploring End-to-End Object Detection with Transformers versus YOLOv8 for Enhanced Citrus Fruit Detection within Trees", Systems and Soft Computing, Vol. 6, 2024, p. 200103.

[21] Y. Luo, X. Cao, J. Zhang, P. Cheng, T. Wang, Q. Feng, "Dynamic Multi-Scale Loss Optimization For Object Detection", Multimedia Tools and Applications, Vol. 82, No. 2, 2022, pp. 2349-2367.

[22] K. G. Panchbhai, M. G. Lanjewar, "Enhancement of tea leaf diseases identification using modified SOTA models", Neural Computing & Applications, Vol 37, 2025, pp. 2435-2453.

[23] K. G. Panchbhai, M. G. Lanjewar, A. V. Naik, "Modified MobileNet with leaky ReLU and LSTM with balancing technique to classify the soil types", Earth Science Informatics, Vol. 18, 2025, p. 77.

[24] S. Ren, K. He, R. Girshick, J. Sun,"Faster R-CNN: "Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, pp. 1137-1149.