A Benchmarking Framework for Fine-grained Multilabel Semantic Segmentation of Mango Fruit Surface Defects

Maria Jeseca C. Baculo [1,2], Conrado Ruiz Jr. [1,3,*], and Oya Aran [4,*]

[1] College of Computing Studies, De La Salle University, Manila, Philippines
[2] College of Computer Science, Don Mariano Marcos Memorial State University, La Union, Philippines
[3] Department of Engineering Interaction, La Salle-Universitat Ramon Llull, Barcelona, Spain
[4] R&D Data Science and AI, Procter & Gamble in Cincinnati, Ohio, USA
Email: maria_jeseca_baculo@dlsu.edu.ph (M.J.C.B.); conrado.ruiz@salle.url.edu (C.R.J.); aranoya@gmail.com (O.A.)
*Corresponding author

***Abstract*—This study presents a benchmarking framework for multilabel semantic segmentation of mango surface defects. It tackles the real-world issue of overlapping and co-occurring symptoms in agricultural images. The curated dataset includes full fruit images and detailed patches with multi-class annotations. This setup captures the biological complexity of natural crop conditions. Three existing segmentation architectures, DeepLabV3+, HRNet-MultilabelSeg, and You Only Look Once with Vision Transformer (YOLOViT)-MultilabelSeg, are compared using a unified evaluation method. The framework uses standardized metrics, including mean Intersection-over-Union (mIoU), F1-Score, precision, recall, and inference speed. DeepLabV3+ achieved the highest accuracy. Meanwhile, HRNet-MultilabelSeg provided real-time performance at 526 Frames Per Second (FPS) while maintaining competitive segmentation quality. This contribution offers a clear and reproducible benchmark, along with a practical roadmap for public, cross-domain validation and unified training goals.**

***Keywords*— deep learning in agriculture, multilabel semantic segmentation, pixel-level defect annotation, smart precision farming, DeepLabV3+, HRNet-MultilabelSeg, You Only Look Once with Vision Transformer (YOLOViT)-MultilabelSeg**

I. Introduction

In agriculture, visual symptoms of plant diseases often overlap and occur together in complicated ways. Most segmentation methods assume that each region contains a single dominant class. This limits their effectiveness in real-world situations. This paper examines multi-class and multilabel segmentation within a unified framework. The goal is to provide accurate and interpretable segmentations for biological imaging and related fields.

Recent advances in multilabel segmentation—encompassing more robust annotation workflows and architectures designed to handle overlap—have significantly improved accuracy and scalability in complex imagery. As a result, multilabel methods are now a practical choice for tasks that demand fine-grained delineation and dependable scene understanding.

Insights from other fields, particularly bioinformatics, further show the promise of multilabel semantic segmentation. For example, biomedical image analysis often involves segmenting multiple overlapping structures within a single image [1–3]. In cases such as mouse brain scans, where it is essential to identify both somata and blood vessels simultaneously, deep convolutional neural networks combined with custom loss functions and enhancement algorithms have demonstrated the ability to segment multiple object classes with a single model accurately. These methods address common issues, such as class imbalance and annotation errors, highlighting the effectiveness of multilabel segmentation in handling complex scenes with overlapping features [4–5].

In contrast, much of the research on fruit defect detection has primarily focused on single-class detection or object-level classification methods. A recent study by Fan *et al.* [6] introduced a semi-supervised deep learning framework for multilabel classification, aiming for precise disease detection in leaves. However, in this study, multilabel classification was employed to identify multiple diseases in a single leaf, while semantic segmentation was utilized to enhance the analysis results.

For fruit defect detection, the mango (Mangifera indica), particularly the Carabao variety, is a key export product in the Philippines. The Philippine National Standard for Fresh Fruit—Mangoes (PNS/BAFPS 13:2004) [7] outlines evaluation criteria and recognizes 24 known defects. These defects include pre-harvest defects, damage caused by pathogens, insects, and animals, and damage resulting from handling. This study focuses on five primary defects from the categories of pathogens, insects, and handling.

Previous research has explored mango defect detection using various deep-learning architectures [8–12]. For

Manuscript received October 27, 2025; revised November 25, 2025; accepted December 16, 2025; published July 16, 2026.

instance, Villanueva *et al.* [13] used the YOLOv8 model on 2,160 training images to classify three mango surface defects. The model achieved 100% accuracy for black spots, 80% accuracy for brown spots, and 83.33% accuracy for mango scabs, resulting in a mean Average Precision (mAP) of 70% at an IoU threshold of 0.50. However, it did not detect different defect types within the same mango image, although it identified multiple instances of the same defect class.

Similarly, Baculo *et al.* [14] evaluated different object detection frameworks for five classes of mango defects. A modified Faster Region-Convolutional Neural Network (R-CNN), which used EfficientNet as its backbone, achieved an mAP of 0.89 at an Intersection-over-Union (IoU) of 0.50 and 0.84 at an IoU of 0.75. This model could identify multiple defect instances across various classes but produced false positives for connected or overlapping defects.

While traditional object detection methods may be effective for specific agricultural tasks, they often lack the spatial accuracy required to analyze fine-grained fruit surface defects. Mango surfaces present a unique challenge due to their complex textures and the presence of various visually similar defect types that often appear together or overlap. To address these issues, this work develops a robust segmentation pipeline that enables per-pixel multilabel prediction and compares various deep learning architectures in their ability to identify fine-grained defect regions. This paper seeks to offer a reproducible benchmarking framework with standardized inputs and outputs, consistent training and evaluation procedures, and synchronized inference timing. It also provides clear guidance on accuracy and throughput using standard metrics (mean Intersection-over-Union (mIoU), F1, precision, recall, and Frames Per Second (FPS)), as well as an analysis of confusion that highlights failure modes.

By leveraging a carefully curated, pixel-level-annotated dataset and recent advancements in convolutional and hybrid transformer architectures, this study examines how segmentation approaches can be enhanced to detect various mango defect types. The insights gained will inform model selection methods, dataset organization, and evaluation protocols specifically designed for agricultural quality assessment.

II. Methodology

Fig. 1 illustrates the study's framework, where each model follows a distinct architecture, training objective, and loss function, yet shares the same input format and annotation structure.

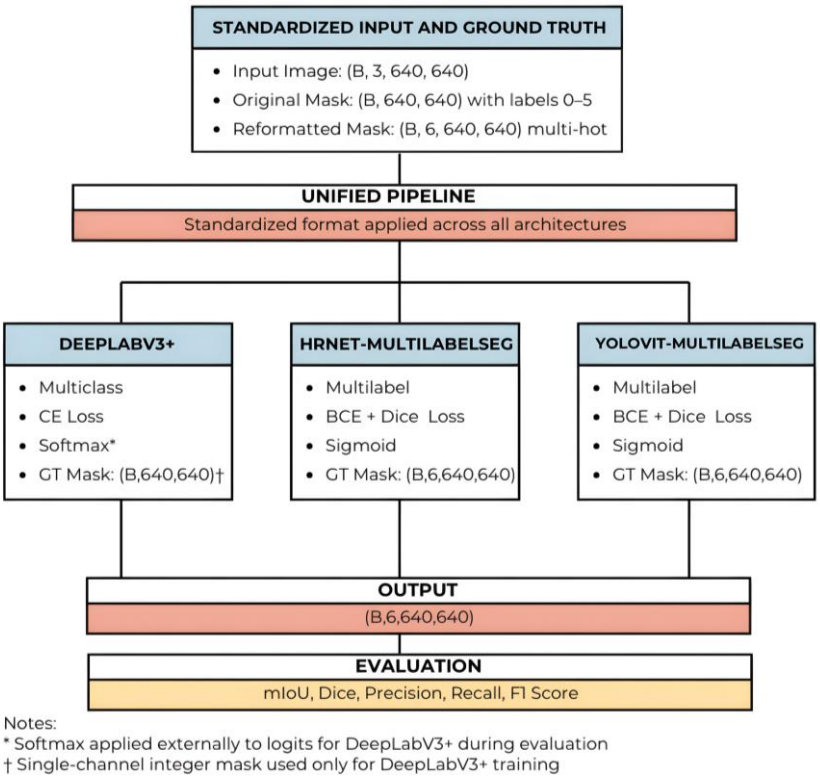


Fig. 1. Benchmarking workflow for mango defect segmentation.

A. Dataset Preparation

The dataset comprises 1,913 mango images annotated with polygonal masks across five defect categories: Anthracnose/Stem-end rot, Cecid, Latex defects, Mealy Bug, and Scab. To reflect real-world complexity, the dataset emphasizes co-occurrence and overlapping defects, with two or more classes present in most samples. Images with a single defect type accounted for approximately one-third of the dataset, achieving a balanced representation of overlapping symptoms. The diverse conditions under which the images were captured are illustrated in Fig. 2.

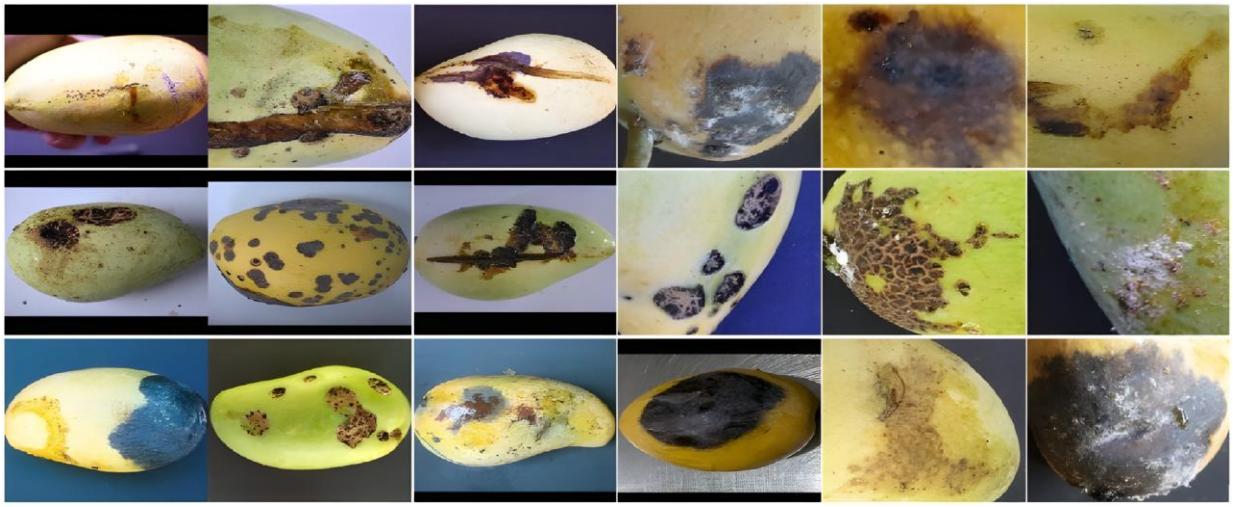


Fig. 2. Representative dataset variability under real capture conditions.

To understand how the annotated defects are distributed across the dataset, Fig. 3 shows a clear imbalance in class representation. Some defect categories are more common than others. The Scab class accounts for the largest share, approximately 58.49% of the total labeled instances. It is followed by Latex defects and Anthracnose/Stem-end rot. In contrast, Cecid and Mealy Bug are less common, contributing a small percentage to the dataset. The complex visual features of mango surface defects create unique challenges that require a segmentation approach going beyond traditional single-label classification.

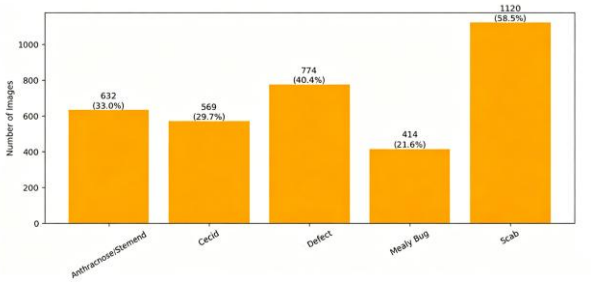


Fig. 3. Number of images containing each defect category.

In many cases, defects are not isolated but occur together on the same fruit. They often overlap or form connected patterns that span multiple categories. The boundaries between defect types can be unclear. Their overlapping and connected nature make it insufficient to assign a single class to each polygon, as seen in standard multi-class segmentation. At times, a multilabel segmentation framework is necessary. This framework enables simultaneous detection of multiple defect categories at the pixel level. As shown in Fig. 4, a large portion of the dataset includes images with multiple annotated defect classes. Most images feature two defect types, along with a significant number of single-defect images.

To gain deeper insights into the relationships among defect classes, we created a co-occurrence matrix to measure the frequency with which pairs of defects appear together in the same image. As shown in Fig. 5, the matrix reveals significant inter-class co-occurrence, highlighting the need for a multilabel segmentation framework. The diagonal elements indicate the total number of images in which each class is present, matching the individual class frequencies. More importantly, the off-diagonal values show how often each pair of classes co-occurs. Figs. 3–5 report per-class frequency, with Scab being the most prevalent at approximately 58.5%. The majority of images have two or more classes, and the co-occurrence matrix shows strong off-diagonal mass, for example, the co-occurrence of Scab with Latex and Cecid, reinforcing the need for multilabel segmentation.

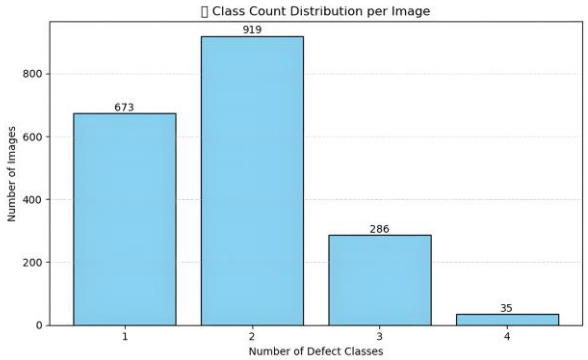


Fig. 4. Distribution of images by the number of defect classes present.

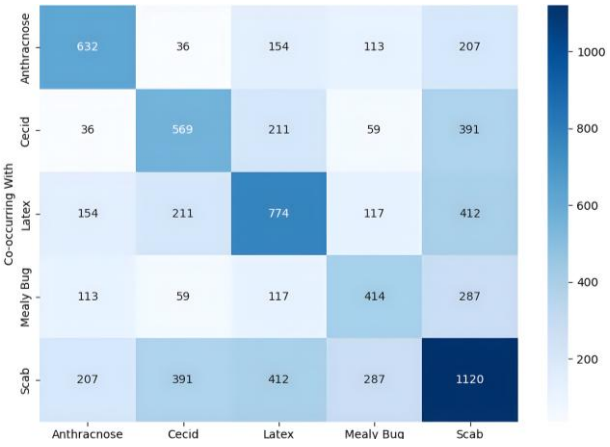


Fig. 5. Pairwise co-occurrence counts for defect classes within the same image.

Notable co-occurrence patterns involve Scab, the most common class, which frequently appears with both Latex defect and Cecid, suggesting potential biological or spatial links. The Latex defect has a high co-occurrence not only with Scab but also with Cecid and Anthracnose/Stem-end, indicating a tendency to co-occur with other defects. The Mealy Bug, while less frequent, is often found alongside Scab and Latex defects.

These patterns highlight that mango surface defects are not mutually exclusive, as multiple pathologies often co-occur. Therefore, single-label classification or segmentation methods would not capture the true complexity of the annotation landscape. Each class-specific annotation was exported as a binary mask, with white pixels representing the defect area and black pixels denoting the background (Fig. 6). These masks were organized in class-wise directories to support structured processing.

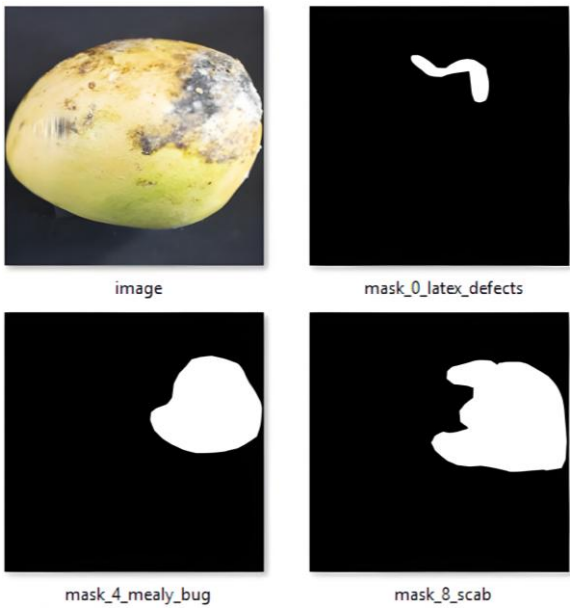


Fig. 6. Example multilabel ground-truth encoding.

Per-class polygon masks are stacked into a 6-channel multi-hot tensor (background + 5 classes) for multilabel models, so multiple channels may be active at a given pixel. For DeepLabV3+, each pixel is assigned a discrete integer label from 0 to 5 (0 for background and 1–5 for defect classes), stored as single-channel PNG masks for cross-entropy training. All ground-truth masks are stored in PNG format for subsequent use in segmentation and detection tasks. To support the specific learning objectives of each architecture, dedicated preprocessing procedures were applied, maintaining the common input resolution of 640×640.

B. Models Benchmarked

The following architectures are benchmarked. DeepLabV3+ is designed for multi-class semantic segmentation [15, 16]. It employs Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual features [17] and uses a ResNet34 backbone for feature extraction. It is trained with CrossEntropyLoss, which assumes a single dominant class per pixel—enforcing mutual exclusivity.

HRNet-MultilabelSeg integrates a high-resolution HRNet CNN backbone with a Transformer block to model global dependencies. Its CrossFusion decoder merges fine-grained convolutional features with global attention representations. The model outputs sigmoid-activated multilabel predictions, allowing for concurrent class assignments at each pixel. It is trained using BCEWithLogitsLoss on 6-channel binary masks, fully supporting spatial overlap among defects. This design enables the model to recognize complex, co-occurring symptoms, which are critical for biological imaging [18, 19].

You Only Look Once with Vision Transformer (YOLOViT)-MultilabelSeg combines a ResNet-18 encoder with a Vision Transformer (ViT) to capture both local and contextual information, followed by a lightweight decoder for pixel-wise predictions. Like HRNet, it supports multilabel segmentation through sigmoid outputs and is trained using BCEWithLogitsLoss on 6-channel binary masks. The key distinction in this work lies in how segmentation is framed: DeepLabV3+ treats it as a mutually exclusive multi-class problem, while HRNet-MultilabelSeg and YOLOViT-MultilabelSeg adopt a multilabel formulation—where a pixel may simultaneously belong to multiple defect classes. This design choice is critical for modeling real-world mango defects, which frequently manifest in overlapping or clustered regions.

The implementation leverages a modular Python-based ecosystem. PyTorch provides the core deep learning framework, while segmentation_models_pytorch is utilized to instantiate the DeepLabV3+ encoder–decoder architecture. While multilabel models use this format directly during training, DeepLabV3+ retains the original single-channel masks for compatibility with its multi-class learning setup. Table I summarizes the architectures compared in this study.

OpenCV and NumPy are used for image preprocessing and tensor conversion, while TorchMetrics handles metric computation and integrates with TensorBoard for monitoring and visualization. The pipeline is intentionally modular, separating data handling, model training, and evaluation so different segmentation families can be swapped in without breaking conventions for inputs, labels, or scoring. Each architecture is trained with targets that match its output semantics, enabling a fair comparison of multilabel capability.

To maintain consistency in comparisons, a unified input–output specification is applied across all models. Ground-truth polygons are converted to class-specific binary masks and, for multilabel models, stacked into a six-channel multi-hot tensor (background plus five defects) so that overlapping regions are explicitly encoded. For DeepLabV3+, a single-channel index mask (0–5) is retained to match its multi-class objective. This scheme preserves compatibility across architectures while capturing the spatial overlap typical of agricultural images in real conditions.

TABLE I. COMPARISON OF DEEPLABV3+, HRNET-MULTILABELSEG, AND YOLOVIT-MULTILABELSEG ARCHITECTURES

Feature	DeepLabV3+	HRNet-MultilabelSeg	YOLOViT-MultilabelSeg
Architecture Type	CNN (Encoder–Decoder)	Hybrid (CNN + Transformer)	Hybrid (CNN + ViT)
Backbone	ResNet34 (ImageNet)	HRNet-style backbone	ResNet18 (custom)
Transformer Module	✗	SimpleTrans × 2	ViT Encoder (4 layers)
Feature Fusion	ASPP + Decoder	CrossFusion (local + global)	Linear projection + ViT + Conv
Upsampling	Bilinear (in decoder)	Bilinear (post-fusion)	Bilinear to 640×640
Segmentation Head	Conv–ReLU–Convs	Conv + ReLU + Dropout + Conv	Conv–ReLU + Conv
Output Activation	**softmax** (multi-class)	**sigmoid** (multilabel)	**sigmoid** (multilabel)
Loss Function	CrossEntropyLoss	BCE + Dice	BCE + Dice
Num. of Classes	6 (5 + background)	6 (5 + background)	6 (5 + background)
Input Size	640×640	640×640	640×640
Output Size	640×640	640×640	640×640
Pretrained Weights	ResNet34 (ImageNet)	ResNet18 (partial)	None (ViT custom)
Parameters (Approx.)	~38M	~22M	~25M

For multilabel models, ground-truth annotations are provided as 6-channel multi-hot masks (background plus five defect classes), while DeepLabV3+ uses single-channel index masks to match its multi-class objective.

C. Handling Class Imbalance

The dataset exhibits a pronounced class imbalance, with Scab accounting for most labeled pixels and Cecid and Mealy Bug occurring much less frequently. To mitigate this effect at least partially, the multilabel models are trained using a compound loss function that combines binary cross-entropy with logits (BCEWithLogitsLoss) and a Dice loss term, applied independently to each defect channel (Fig. 7). Model outputs are produced as logits; these logits are passed directly to BCEWithLogitsLoss, which compares them with the binary ground-truth masks. In parallel, the logits are transformed by a sigmoid activation to obtain per-pixel, per-class probabilities, which are used to compute the Dice loss.

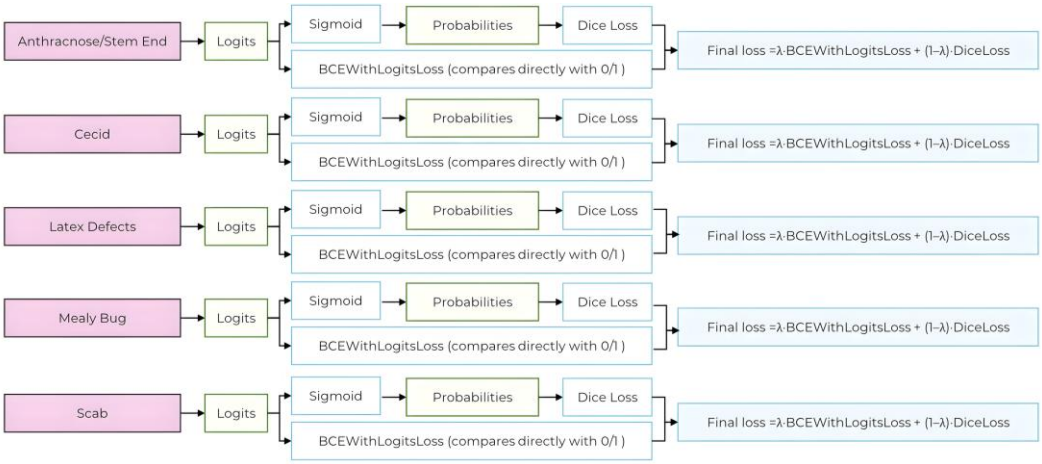


Fig. 7. Multilabel loss computation used for class-wise supervision.

The BCE component promotes calibrated per-pixel predictions, whereas the Dice component is overlap-based and normalizes the intersection by the union of predicted and ground-truth masks, implicitly increasing the relative contribution of small structures and rare classes to the gradient signal. In practice, this combination provides a "soft" handling of imbalance that helps stabilize training and reduces the tendency of the optimization to be driven primarily by majority-class pixels.

More explicit imbalance-aware strategies, such as class-weighted BCE or focal loss, are not employed in the current benchmark. As reflected in the per-class metrics, rare classes such as Cecid and Mealy Bug still lag behind Scab in segmentation quality. More aggressive imbalance mitigation—through learned class weights, focal variants, or targeted augmentation of underrepresented defects—therefore remains an important direction for future extensions of this framework.

D. Performance Metrics

To assess and compare the performance of the three models in mango defect segmentation, a consistent set of evaluation metrics was employed. All models are configured to output pixel-wise segmentation maps, either as single-channel class indices (for multi-class classification) or multi-channel binary masks (for multilabel classification). The unified dataset structure, standardized input size, and consistent annotation formats allow for a reliable, architecture-agnostic comparison [19]. Subsequent evaluation procedures compute segmentation quality across all models using a fixed set of metrics.

Four primary metrics were used to evaluate segmentation quality: mean Intersection-over-Union

(mIoU), per-class F1-Score, per-class precision, and per-class recall. These metrics are suitable for evaluating segmentation models in multilabel settings, especially when defect regions may overlap, and class imbalance is present.

Mean Intersection-over-Union (mIoU) serves as the principal metric. It measures the overlap between predicted and ground truth regions per class and averages these values across all defect classes. By accounting for both false positives and false negatives, mIoU provides a comprehensive measure of segmentation accuracy. Formally, for each class *c*, the Intersection over Union is defined as:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}$$

where TP_c, FP_c, and FN_c, represent true positives, false positives, and false negatives for class *c*, respectively. The *mIoU* across all *C* classes is then:

$$Mean\ IoU = \frac{1}{C}\sum_{c}^{C} IoU_c$$

Per-class F1 is the harmonic mean of precision and recall, providing a single measure that balances sensitivity and specificity. It is instrumental in multilabel segmentation, where defects often overlap and vary in size. For class *c*, the F1-Score is calculated as:

$$F1_c = 2 \times \frac{Precision_c \times Recall_c}{Precision_c + Recall_c}$$

Per-class precision indicates the proportion of correctly predicted pixels among all pixels predicted as belonging to class *c*, reflecting the model's ability to avoid over-segmentation:

$$Precision_c = \frac{TP_c}{TP_c + FP_c}$$

Per-class recall measures the proportion of actual pixels of class *c* correctly identified by the model, showing the effectiveness in detecting true defect regions:

$$Recall_c = \frac{TP_c}{TP_c + FN_c}$$

Applying these metrics consistently across the three models ensures a reliable and equitable comparison of their segmentation performances. The per-class analysis further highlights each model's strengths and weaknesses concerning the five defect categories, guiding future model refinement and dataset balancing efforts.

To further assess model performance and error distribution across defect categories, a normalized confusion matrix is generated. This matrix visualizes the proportion of predictions per class, normalized by the total number of actual instances in each ground truth category. It provides insight into inter-class misclassification tendencies and supports qualitative interpretation of segmentation reliability.

Inference speed and Frames Per Second (FPS) quantify how efficiently a model performs during prediction. In this study, these metrics were obtained by isolating and timing only the model's forward pass, excluding any additional time for data loading, preprocessing, or post-processing. This procedure provides a focused measurement of the model's pure inference capability.

To make the reported frame rates interpretable and comparable, all inference experiments were conducted on a single workstation with the hardware and software configuration summarized in Table II. The models were evaluated on an NVIDIA GeForce RTX 4060 GPU, paired with an Intel 64 Family 6 Model 158 CPU (4 logical/4 physical cores) and 25.7 GB of system RAM, running Windows 10. The software stack consisted of Python 3.10.18, PyTorch 2.2.2 with CUDA 12.1 and cuDNN 8801, together with standard vision libraries such as OpenCV, NumPy, scikit-learn, and Torchvision.

TABLE II. HARDWARE AND SOFTWARE CONFIGURATION OF THE WORKSTATION USED TO MEASURE SINGLE-IMAGE INFERENCE

Component	Specification
GPU	NVIDIA GeForce RTX 4060
CPU	Intel64 Family 6 Model 158 Stepping 11
CPU cores	4 logical / 4 physical
RAM	25.7 GB
OS	Windows 10 (10.0.26100)
Python	3.10.18
PyTorch	2.2.2+cu121
CUDA (from PyTorch)	12.1
cuDNN	8801
Key libraries	OpenCV; NumPy; scikit-learn; Torchvision

All FPS values reported in this paper correspond to single-image inference with a batch size of 1 and an input resolution of 640×640 pixels. During timing, each model was placed in evaluation mode and run under torch.no_grad() to disable gradient computation. Only the forward pass on the Graphics Processing Unit (GPU) was measured; data loading, preprocessing, and metric computation were not included. To obtain stable estimates and avoid optimistic timings due to asynchronous Compute Unified Device Architecture (CUDA) execution, several warm-up iterations were discarded, and torch.cuda.synchronize() was called immediately before starting and after stopping the timer. The mean per-image latency over the remaining iterations was then inverted to obtain FPS.

GPU timing was therefore based on explicit synchronization to avoid under-reporting. Because CUDA kernels execute asynchronously, wall-clock timers around a forward pass can otherwise measure only kernel dispatch time. By bracketing the timed block with torch.cuda.synchronize(), all queued kernels were forced to complete, yielding the actual inference latency. Reporting both latency and FPS in this way provides a transparent view of prediction efficiency and supports fair

comparison with other real-time or resource-constrained deployments.

III. Results

A. Quantitative Results

The evaluation of three segmentation architectures—DeepLabV3+, HRNet-MultilabelSeg, and YOLOViT-MultilabelSeg—shows different levels of effectiveness in locating and identifying multiple, often overlapping, mango surface defects. All three models were trained on the dataset without any augmentation. The dataset was split into 90% (1724 images) for training and 10% (191 images) for testing. The models were trained for 300 epochs, beginning with a learning rate of 0.002 and a dropout rate of 0.1.

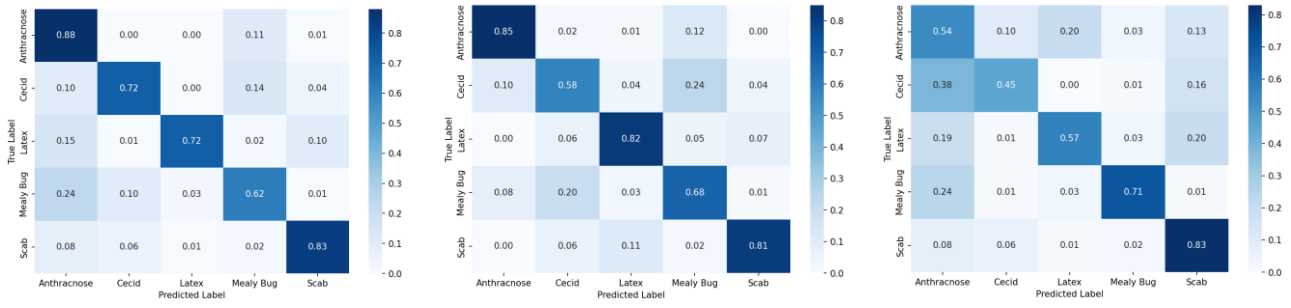
Among the models tested, DeepLabV3+ yielded the most reliable and robust results on key segmentation metrics. It reached a mIoU of 0.652 and a mean F1-Score of 0.756. This indicates sound boundary localization and a high overlap with actual defect areas across defect classes. HRNet-MultilabelSeg performs similarly well, with mean F1 only slightly below DeepLabV3+. Its per-class scores indicate effective handling of detailed and clustered defect patterns. However, performance varies by class, particularly for defects like Cecid.

Direct numerical comparison with existing work is not trivial, since most recent studies on mango quality assessment focus on classification or object detection and report accuracy or mean Average Precision (mAP) rather than pixel-wise mIoU. For example, Villanueva *et al.* [12] report an mAP of 70% at IoU 0.50 for YOLOv8-based detection of three defect types, and Baculo *et al.* [13] obtain mAP values of 0.89 and 0.84 at IoU thresholds of 0.50 and 0.75, respectively, using a Faster R-CNN variant for five defect classes. These systems operate at the object level and do not explicitly model overlapping defects at the pixel scale. In contrast, the present benchmark addresses a denser and more demanding multilabel segmentation setting, where multiple defect classes may co-occur within the same local region.

When viewed alongside recent DeepLabV3+-style segmentation approaches in other high-resolution domains, such as remote sensing and medical imaging, which also report strong but dataset-dependent mIoU values on complex scenes [14, 15], an mIoU of 0.652 for DeepLabV3+ can be regarded as a competitive baseline for fine-grained, multilabel fruit defect segmentation. This level of performance shown in Table III provides a realistic starting point for subsequent architectural improvements and dataset extensions targeting even more challenging agricultural scenarios.

TABLE III. Comparison of Segmentation Models across Performance and Efficiency Metrics

Model	mIoU	Precision	Recall	F1-Score	Time (s)	FPS
DeepLabV3+	0.652	0.798	0.754	0.756	0.0459	21.8
HRNet-MultilabelSeg	0.52	0.75	0.748	0.749	0.0019	526.43
YOLOViT-MultilabelSeg	0.416	0.658	0.62	0.624	0.0124	80.6



(a) (b) (c)

Fig. 8. Normalized confusion matrices for the three benchmarked models. (a) DeepLabV3+; (b) HRNet-MultilabelSeg; (c) YOLOViT-MultilabelSeg.

The three models illustrate a clear speed-complexity trade-off. DeepLabV3+ uses a ResNet-34 encoder, a decoder, and an ASPP to achieve reliable segmentation. It takes about 0.0459 s per image (approximately 21.8 FPS), making it better suited to offline or batch processing than to strict real-time applications. On the other hand, HRNet-MultilabelSeg keeps high-resolution features with lightweight fusion and transformer blocks. It processes an image in about 0.0019 s (approximately 526 FPS) while maintaining spatial detail. This makes it an excellent choice for real-time applications. YOLOViT-MultilabelSeg sits between the two at approximately 0.0124 s per image (~80.6 FPS); inference is efficient, although overlap quality lags, likely due to simpler upsampling and fewer heads. In practice, the choice of architecture should prioritize latency budgets and hardware limits over raw accuracy.

HRNet-MultilabelSeg achieves the fastest inference but shows segmentation quality that is noticeably better than YOLOViT-MultilabelSeg, though not as sharp as DeepLabV3+'s. Some predictions still indicate slight fragmentation or soft boundaries, particularly in complex spatial layouts. In some cases, the models appear to segment more extensive regions than annotated in the ground truth, which may reflect under-annotation in multilabel areas rather than actual over-segmentation. The architecture's high-resolution pathways enable it to retain important spatial features, resulting in a segmentation

performance that strikes a balance between speed and accuracy [20].

To further understand model behavior, normalized confusion matrices (Fig. 8) were generated for each model. These matrices offer insight into specific misclassification tendencies, particularly in multilabel situations where classes often overlap within the same image. DeepLabV3+ exhibits minimal cross-class confusion, indicating distinct class boundaries and robust discriminative learning. In contrast, the YOLOViT-MultilabelSeg model exhibits greater dispersion among its off-diagonal entries. This means difficulties in separating visually overlapping defects, such as Anthracnose/Stem-End, Cecid, and Latex defects.

B. Ablation Study on HRNet-MultilabelSeg

To assess the contribution of key architectural components in HRNet-MultilabelSeg, an ablation study was conducted using three progressively simplified configurations (Table IV). The baseline model, which includes both CrossFusion and the complete set of Transformer blocks, attains the best overall performance with an IoU of 0.520, precision of 0.750, recall of 0.748, and an F1-Score of 0.749 at 526.43 FPS. This configuration can therefore be regarded as the reference design, providing the highest segmentation quality while already operating at real-time speed.

TABLE IV. COMPARISON OF HRNET-MULTILABELSEG VARIANTS ACROSS PERFORMANCE AND EFFICIENCY METRICS

Model Variant	MIoU	Precision	Recall	F1	FPS
baseline	0.520	0.750	0.748	0.749	526.43
no_fuse	0.482	0.709	0.652	0.679	509.33
no_trans	0.456	0.716	0.649	0.681	650.00
1_simpletrans	0.185	0.508	0.318	0.391	558.98

Removing the fusion module (“no_fuse”) results in a noticeable drop in mIoU from 0.520 to 0.482 and a decrease in F1 from 0.749 to 0.679, with both precision and recall also reduced. The FPS remains in the same range (509.33), indicating that fusion contributes substantially to mask quality but has only a minor impact on throughput. This suggests that multi-scale feature fusion plays an important role in refining the spatial support of the defect regions, particularly for fine structures and boundaries.

In the “no_trans” configuration, the Transformer blocks are removed while the fusion pathway is preserved. mIoU decreases further to 0.456, whereas precision and recall remain close to those of “no_fuse,” yielding a similar F1-Score (0.681). In contrast, inference speed increases to 650 FPS. This pattern indicates that the Transformer modules mainly enhance spatial coverage and boundary consistency—reflected in mIoU—while the overall pixel-wise class decisions remain broadly similar. The associated computation is non-negligible, so discarding the Transformers may be attractive in scenarios where latency constraints are more stringent than the requirement for fine-grained delineation.

The most aggressive simplification (“1_simpletrans”) results in pronounced degradation across all metrics, with mIoU falling to 0.185 and F1 to 0.391, while maintaining real-time processing at 558.98 FPS. The large gap between this configuration and the baseline implies that severely reducing the capacity of the high-resolution and Transformer pathways prevents the network from forming sufficiently expressive representations of the multilabel defect patterns. In this setting, small lesions are frequently missed, and overlapping defects are poorly separated.

Overall, the ablation study indicates that both the high-resolution fusion pathway and the Transformer blocks contribute meaningfully to multilabel segmentation quality while still allowing real-time inference. Moderate simplifications (“no_fuse” or “no_trans”) yield substantial gains in throughput only when Transformers are removed, but at the cost of reduced mIoU. More aggressive reductions in capacity quickly compromise segmentation fidelity. For applications that prioritize accuracy, the baseline configuration remains the most suitable choice, whereas the “no_trans” variant offers a pragmatic alternative when extremely high frame rates are required.

These observations suggest a natural next step in model design. A future variant could replace the custom HRNet-style backbone with a truncated ResNet-18. In such a design, the deepest feature map would be projected to a fixed-width representation and passed through a small stack of Transformer blocks. This branch would provide global context in a more explicit and hardware-friendly way than a full multi-branch HRNet. The global representation could then be fused with the local convolutional features via a residual fusion layer, while a higher-resolution skip connection from an earlier ResNet stage provides fine spatial detail. A compact decoder with only a few convolutional layers and a single bilinear upsampling step would complete the network, keeping parameter count and latency low.

Such modifications are important for several reasons. First, grounding the encoder in a widely used ResNet backbone would make the model easier to reproduce, extend, and integrate into existing computer-vision pipelines, since pretrained weights and implementation details are already well established in the community. Second, separating local convolutional processing, global transformer reasoning, and cross-fusion into clearly defined modules creates a more interpretable architecture: ablations can be mapped directly onto individual components, and practitioners can adjust capacity or depth in each branch according to their computational budget. Third, a lighter backbone and decoder, together with carefully designed attention blocks, are likely to yield a more favorable accuracy-throughput trade-off for embedded or on-device deployment, which is particularly

relevant for real-time grading of fruit at the point of harvest or in packing facilities. Finally, this modular design would make it easier to adapt the framework to other crops or defect types, since only the dataset-specific heads and training configuration would need to change, while the core architectural blocks remain reusable.

C. Representative Predictions and Error Patterns

Since the mIoU of YOLOViT-MultilabelSeg fell below 0.50, a commonly used reference threshold for acceptable segmentation performance, the sample predictions of DeepLabV3+ and HRNet-MultilabelSeg are presented. In the qualitative examples shown in Fig. 9, both DeepLabV3+ and HRNet-MultilabelSeg produce segmentation maps that closely follow the manually annotated ground truth. Across the rows, the main defect regions are captured with correct class labels and contours that align well with the visual boundaries in the input images. Large, contiguous lesions are reproduced with coherent shapes, and the models respect the separation between different defect types even when they are spatially close.

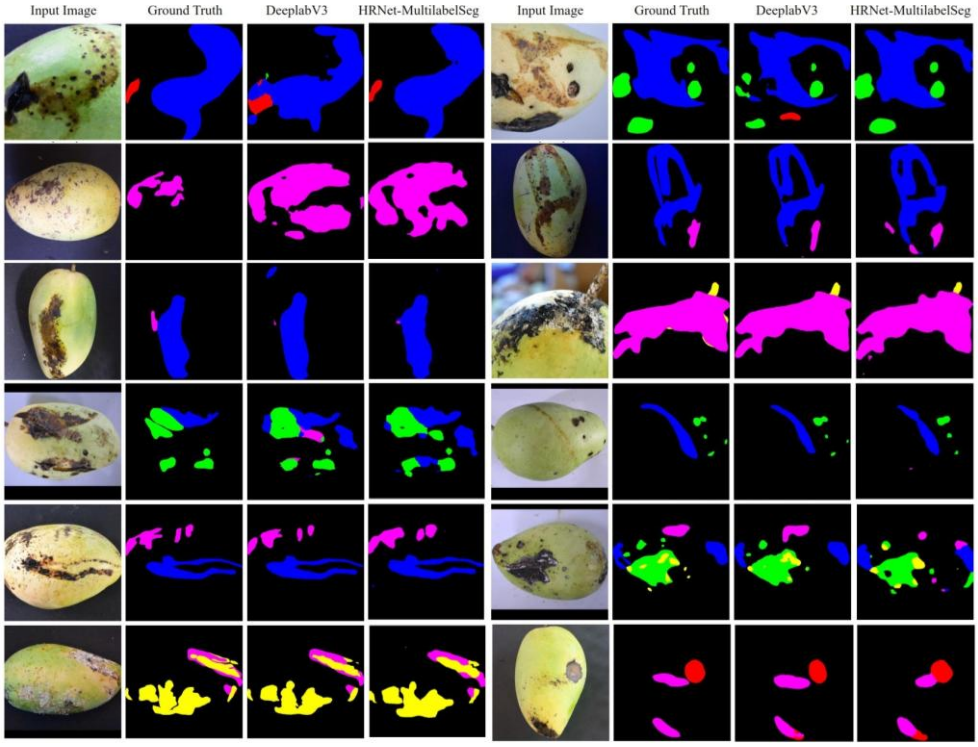


Fig. 9. Qualitative examples of successful multilabel segmentation.

Several patterns are evident when comparing the two models. DeepLabV3+ tends to generate slightly thicker masks, sometimes extending by a few pixels beyond the annotated borders, making the predictions appear more "filled in". HRNet-MultilabelSeg generally produces sharper, slightly more compact regions, particularly for thin, elongated defects on the fruit surface. In the rows where multiple classes co-occur, both models preserve the multilabel structure: overlapping or adjacent defects are assigned consistent class colors, and there is little evidence of label bleeding between categories.

The examples with many small scattered lesions further highlight the trade-off between sensitivity and precision. DeepLabV3+ often retains more tiny specks, matching the ground truth but occasionally introducing additional tiny fragments. HRNet-MultilabelSeg, by contrast, is slightly more conservative, suppressing some of the smallest regions while maintaining clean, well-formed masks for the dominant lesions. The final row illustrates near-perfect delineation of isolated circular and elongated defects by both architectures, with HRNet-MultilabelSeg producing particularly smooth boundaries. Overall, these qualitative results support the quantitative findings: DeepLabV3+ offers the highest overall accuracy, while HRNet-MultilabelSeg delivers visually crisp segmentations with competitive performance, especially in multilabel scenes.

Fig. 10 illustrates representative failure cases observed in the test set, highlighting scenarios where the models' predictions deviate from the ground truth. DeepLabV3+ achieves the highest segmentation accuracy in complex scenes, but in minor defects, it can be observed that HRNet-MultilabelSeg can distinguish more patterns at the pixel-level. HRNet-MultilabelSeg offers a solid middle ground between precision and speed. These results indicate that newer hybrid architectures offer promising options; however, traditional encoder-decoder networks, such as DeepLabV3+ and HRNet-MultilabelSeg, remain more reliable for complex multilabel segmentation in agriculture. Adding additional supervision, class-balancing methods, or attention-focused decoders could help close this performance gap in future versions.

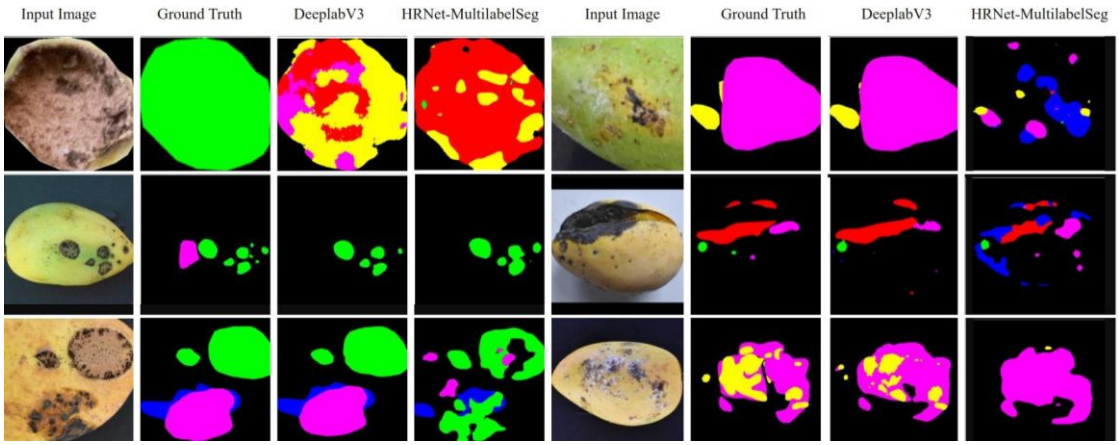


Fig. 10. Qualitative failure cases and typical error modes.

The second left-hand example contains several small, circular lesions of two different classes (Scab and Cecid defect in the ground truth). Here, both models recover the approximate positions and shapes of the spots, but almost all are predicted as Cecid defects. This indicates that when defects are small and visually similar in color and texture, the networks tend to collapse the multilabel distinction and treat them as a single category, even though the binary "defect versus background" decision is mainly correct.

The third left-hand example shows larger lesions with overlapping classes. DeepLabV3+ produces relatively clean blobs that follow the gross shape of the annotations but smooths out some internal boundaries. HRNet-MultilabelSeg, by contrast, over-segments the same area into many small fragments and introduces additional classes inside the main lesion. In the lower row, HRNet-MultilabelSeg generates a mixture of patches, with the ground truth containing primarily two compact regions. This suggests that HRNet-MultilabelSeg is sometimes overly sensitive to local texture variations, splitting a single annotated lesion into several class predictions.

The right half of the figure illustrates similar failure modes in different scenes. In the top example, the ground truth consists of one large Scab defect with a smaller Mealy Bug at the side. DeepLabV3+ reproduces this structure reasonably well, whereas HRNet-MultilabelSeg breaks the same area into scattered regions, again revealing a tendency to fragment contiguous lesions and to confuse nearby classes.

Overall, these qualitative failure cases show that both architectures can correctly identify defective fruit, but they differ in how they partition complex or fine-scale patterns. DeepLabV3+ tends to be more consistent at the class level but may oversmooth boundaries or add small spurious regions, whereas HRNet-MultilabelSeg can either over-fragment a lesion into multiple classes or merge distinct small defects into a single region. These patterns complement the quantitative results by illustrating the kinds of errors that remain challenging in multilabel, overlapping defect segmentation.

IV. Conclusion and Future Works

This study presents a pipeline for multilabel, pixel-level segmentation of mango surface defects, with a particular focus on overlapping and co-occurring symptoms. A carefully annotated dataset of 1913 images across five defect classes—Anthracnose/Stem-end, Scab, Latex defects, Mealy Bug, and Cecid—supports fair and reproducible evaluation under realistic agricultural conditions. Three segmentation models are examined: DeepLabV3+, HRNet-MultilabelSeg, and YOLOViT-MultilabelSeg. All models share consistent preprocessing, annotation formats, and evaluation metrics, including mIoU, per-class precision, recall, and F1-Scores, enabling an equitable comparison across architectures and addressing a notable gap in agricultural computer vision.

The results indicate that DeepLabV3+ currently provides the highest overall segmentation accuracy, while HRNet-MultilabelSeg offers a favorable balance between accuracy and latency, making it a credible candidate for real-time quality control settings. Data limitations temper this strength: minimal augmentation and the sparse representation of Mealy Bug and Cecid likely suppress recall for these classes. Performance is expected to benefit from more aggressive imbalance-aware strategies, such as class-balanced augmentation (e.g., targeted oversampling or lesion copy-pasting), synthetic sample generation, and loss reweighting or focal variants. The benchmarking framework itself is model-agnostic and modular, suggesting that the methodology can be extended to other crops and quality inspection systems with relatively minor adaptations.

The ablation study further highlights that HRNet-MultilabelSeg can be improved architecturally. The observed contributions of the fusion pathway and Transformer blocks suggest promising future variants that more explicitly separate local convolutional encoding, global self-attention, and cross-fusion, for example, by adopting a truncated ResNet encoder with lightweight Transformer modules and a compact decoder. Such designs could retain high-resolution detail while offering even better control over the trade-off between segmentation fidelity and throughput, particularly for deployment on constrained hardware.

Beyond these specific refinements, future work may explore richer attention mechanisms, hybrid decoders, and additional contextual cues, such as temporal sequences or multi-view imagery, to better handle complex defect

patterns and challenging backgrounds. For deployment, key challenges include maintaining efficiency on limited hardware, ensuring robustness across orchards, cultivars, and lighting conditions, and producing outputs that are interpretable for line operators and field technicians. Incorporating uncertainty estimation and lighter model variants may further support post-harvest workflows and in-field decision-making. Overall, the presented benchmark establishes a structured foundation for multilabel defect segmentation in agricultural imagery and provides a basis for developing reliable and practically useful solutions in precision agriculture.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Maria Jeseca C. Baculo conducted the experimentation, implemented the proposed methods, performed data analysis and evaluation. Dr. Conrado Ruiz Jr. and Dr. Oya Aran contributed to the conceptualization of the study, provided technical guidance on model design and methodology, and assisted in the interpretation of results. All authors reviewed and approved the final version of the manuscript.

REFERENCES

[1] A. N. Omeroglu, H. M. A. Mohammed, E. A. Oral, and S. Aydin, "A novel soft attention-based multi-modal deep learning framework for multilabel skin lesion classification," *Engineering Applications of Artificial Intelligence*, vol. 120, 105897, 2023. doi: 10.1016/j.engappai.2023.105897

[2] R. Widyaningrum, I. Candradewi, N. R. A. S. Aji, and R. Aulianisa, "Comparison of multilabel U-Net and mask R-CNN for panoramic radiograph segmentation to detect periodontitis," *Imaging Science in Dentistry*, vol. 52, no. 4, pp. 383–392, 2022. doi: 10.5624/isd.20220032

[3] X. Liu, S. Wang, J. C.-W. Lin, and S. Liu, "An algorithm for overlapping chromosome segmentation based on region selection," *Neural Computing and Applications*, vol. 36, no. 1, pp. 133–142, 2024. doi: 10.1007/s00521-023-08646-8

[4] P. Bevandić, M. Oršić, I. Grubišić, J. Šarić, and S. Šegvić, "Multi-domain semantic segmentation with overlapping labels," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV'22)*, 2022, pp. 2615–2624.

[5] X. Wu, Y. Tao, and G. He *et al.*, "Boosting multilabel semantic segmentation for somata and vessels in the mouse brain," *Frontiers in Neuroscience*, vol. 15, 610122, 2021. doi: 10.3389/fnins.2021.610122

[6] K.-J. Fan, B.-Y. Liu, W.-H. Su, and Y. Peng, "Semi-supervised deep learning framework based on a modified pyramid scene parsing network for multilabel fine-grained classification and diagnosis of apple leaf diseases," *Engineering Applications of Artificial Intelligence*, vol. 151, 110743, 2025. doi: 10.1016/j.engappai.2025.110743

[7] Philippine National Standard for Fresh Fruits, Mangoes PNS/BAFPS 13:2004.

[8] B. Li, C. Yao, C.-T. Su, J.-P. Zou, J. Wu, N. Chen, and Y.-D. Liu, "Detection of skin defects on mangoes based on hyperspectral imaging combined with band ratio and improved Otsu method," *Microchemical Journal*, vol. 197, 109718, 2024. doi: 10.1016/j.microc.2024.109718

[9] M. S. R. Kona, A. Guvvala, V. V. L. Eedara, M. S. Gowri, and V. Aluri, "Mango fruit defect detection using MobileNetV2," in *Proc. 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP'24)*, 2024, pp. 22–27.

[10] S. Jadhav and J. Singh, "Design of EGTBoost classifier for automated external skin defect detection in mango fruit," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 47049–47068, 2024. doi: 10.1007/s11042-024-18642-y

[11] A. Gaikwad, M. Deshpande, and V. Bhole, "Fruits defect detection: A review with image processing techniques," *Responsible Production and Consumption*, pp. 221–228, 2024.

[12] S. S. Sekaran and M. Subaji, "Optimized YOLO and Semi-Supervised Fuzzy Graph Convolutional Network (SSFGCN) for surface defect recognition," *Journal of Advances in Information Technology*, vol. 15, no. 12, pp. 1355–1365, 2024.

[13] K. K. Villanueva, J. A. M. Galindo, A. J. R. Tamayo, J. E. C. Rosal, and D. I. E. Hisola, "Development of a computer vision application for mango (Mangifera indica L.) fruit defect detection using YOLOv8 architecture," in *Proc. 2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET'24)*, 2024, pp. 483–487. doi: 10.1109/IICAIET62352.2024.10730444

[14] M. J. C. Baculo, F. P. Cuison, and N. R. D. V. Rivera, "Image-based Mangifera indica pathogen recognition using artificial intelligence," in *Proc. 2024 10th International Conference on Computer Technology Applications (ICCTA'24)*, 2024, pp. 157–161. doi: 10.1145/3665526.3665613

[15] Y. Wang, L. Yang, X. Liu, and P. Yan, "An improved semantic segmentation algorithm for high-resolution remote sensing images based on DeepLabv3+," *Scientific Reports*, vol. 14, no. 1, 9716, 2024. doi: 10.1038/s41598-024-56236-3

[16] I. Ahmad, J. Amin, M. I. U. Lali, F. Abbas, and M. I. Sharif, "A novel DeepLabv3+ and vision-based transformer model for segmentation and classification of skin lesions," *Biomedical Signal Processing and Control*, vol. 92, 106084, 2024. doi: 10.1016/j.bspc.2023.106084

[17] L.-C. Chen *et al.*, "Rethinking atrous convolution for semantic image segmentation," arXiv preprint, arXiv:1706.05587, 2017.

[18] C. Feng, R. Zhang, and L. Guo, "HR-xNet: A novel high-resolution network for human pose estimation with low resource consumption," in *Proc. 2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG'24)*, 2024, pp. 1–7. doi: 10.1109/FG57751.2024.00014

[19] Z. Wang and M. Blaschko. (2025). On neural architectures, loss functions, evaluation metrics for semantic segmentation. [Online]. Available: https://kuleuven.limo.libis.be/discovery/search?query=any,contains,LIRIAS4205216&tab=LIRIAS&search_scope=lirias_profile&vid=32KUL_KUL:Lirias&offset=0

[20] T. T. Phuong, D. D. Tin, and L. H. Trang, "Improving image representation for surface defect recognition with small data," *Journal of Advances in Information Technology*, vol. 15, no. 5, pp. 572–579, 2024.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).