EAI Endorsed Transactions
on Pervasive Health and Technology
Research Article EAI.EU

FFIS: Design and Development of a Domestic Fruit Freshness Recognition System

Xiayu Liu[1], Zhu Ke[2,3], Yuanhao Ding[1], Yingshan Tang[1] and Yuqian Song[1]

[1]Hexiangning College of Art and Design, Zhongkai University of Agriculture and Engineering, No. 24, Dongsha Street, Haizhu District, Guangzhou City, Guangdong Province, China
[2]People's Government of Jiangdong Town, No. 1 Fuqian Street, Jiangdong Town, Chaoan District, Chaozhou City, Guangdong Province, China
[3]Jiangdong Town Industrial Development Service Center, Chaoan District, Chaozhou City, Guangdong Province, China

Abstract

INTRODUCTION: In everyday households, fruits are often stored for too long, forgotten, or discarded early due to uncertainty about their freshness, leading to avoidable waste, unnecessary cost, and suboptimal dietary habits. A reliable and user-centered fresh produce assessment tool can help people establish a healthier and more sustainable lifestyle. The existing freshness assessment systems are mainly designed for industrial or laboratory environments and rarely meet the privacy, stability and cost requirements of family digital health products.
OBJECTIVES: This study aims to design and evaluate an edge-AI system that jointly recognizes fruit category and ordinal freshness stage in real kitchens, providing a reproducible benchmark for household-oriented freshness sensing rather than a one-off engineering prototype.
METHODS: We develop the Fruit Freshness Identification System (FFIS), a lightweight multi-task detector built on YOLO11n with a BiFPN neck, ACmix hybrid attention, and an IoU-based localization loss reweighted for partial occlusion. A kitchen-scene dataset is collected and stratified by household, countertop material, lighting, and clutter. The system is trained with an ordinal regression head for freshness staging and evaluated on COCO-style detection metrics, stage-wise classification metrics, and edge-device throughput and energy consumption.
RESULTS: On the FFIS-Fruit test split, FFIS achieves an mAP@0.5:0.95 of 62.4 ± 0.4, mAP@0.5 of 94.1 ± 0.3, and recall of 90.8 ± 0.3, outperforming lightweight YOLO baselines as well as a two-stage detector-plus-classifier pipeline. Warm-LED and glossy-countertop subsets show consistent gains in high-reflection scenarios. On-device experiments reach ~55 FPS (INT8) on Jetson Orin Nano and ~10 FPS on Raspberry Pi 5 under a unified evaluation protocol.
CONCLUSION: FFIS provides a household-oriented, privacy-preserving reference implementation for kitchen fruit monitoring, demonstrating that single-pass multi-task detection can meet domestic deployment constraints on low-cost hardware. Rather than directly claiming reductions in household food waste, the findings establish a technical foundation for integrating freshness signals into downstream digital-health workflows such as waste-aware reminders and dietary planning.

Keywords: Digital health; Edge AI; Fruit freshness detection; Multi-task learning; On-device inference; Energy efficiency. Received on

22 November 2025, accepted on 6 January 2026, published on 16 January 2026

doi: 10.4108/eetpht.11.11042

*Corresponding author. Email: 1009055089@qq.com

1. Introduction

Food waste is a growing global concern, and household disposal accounts for a substantial share of overall losses [1]. Fruits are often discarded prematurely because of improper storage or misjudged freshness, which not only creates

environmental and economic waste but also disrupts meal planning and dietary habits [2]. Such avoidable disposal is closely related to household diet models, and can be mitigated by healthy-diet-oriented green design approaches that explicitly account for environmental impact and sustainable development [3]. Household kitchens—where most food handling occurs—pose unique challenges for fresh-food management: subtle visual cues (e.g., color and texture) are hard to interpret under varying lighting, countertop reflections, clutter, and partial occlusions. These factors make it difficult for consumers to decide whether fruits are still suitable for consumption, leading to unnecessary waste. From a digital-health perspective, preventing premature disposal supports proactive, user-centered dietary management in everyday homes.

Existing freshness-assessment approaches are largely industrial or laboratory-oriented and require controlled conditions that are impractical for everyday households. Consumer applications typically offer coarse labels (e.g., “ripe”/“unripe”) and lack reliable, fine-grained cues that are robust to real-kitchen environments, leaving users uncertain and encouraging premature disposal.

1.1 Motivation and positioning within digital health

Global health services are shifting from treatment to prevention and from hospital-centric to user-centered models[4]. Within this agenda, dietary behaviors and domestic food handling are modifiable factors with outsized impact but remain poorly instrumented in everyday homes. We therefore frame kitchen fruit monitoring as a digital-health product component: an edge-AI capability that passively senses fruit status and provides timely signals for downstream workflows—e.g., waste-aware reminders, nutrition dashboards, or IoT prompts in smart kitchens. To make these downstream dashboards and reminders usable across different household groups, inclusive design principles should be adopted to address diverse user needs and equity, which can further improve user satisfaction through interdisciplinary collaboration [5].Unlike lab-centric vision models, our focus is on product constraints crucial to digital health: privacy-preserving on-device inference, data minimization, robustness to kitchen-specific artefacts (glare/occlusion), and a reproducible evaluation protocol that supports real-world deployment.

This paper presents the Fruit Freshness Identification System (FFIS), a vision-based, edge-first system that detects both fruit class and ordinal freshness stage in typical kitchen settings. FFIS is privacy-preserving and runs in real time on inexpensive on-device hardware, addressing the practical challenges of variable illumination, specular reflections, clutter, and occlusions.

Our work focuses on a single-pass multi-task detector (YOLO11n + BiFPN + ACmix with an improved IoU-based localization loss), a kitchen-specific dataset with a unified evaluation protocol, and on-device deployment evidence demonstrating real-time or near-real-time performance.

This work is positioned as a digital-health–oriented, household reference implementation for domestic fruit monitoring, rather than a radically new detection paradigm. In home digital health ecosystems, everyday dietary choices and food-handling behaviors are key modifiable factors; however, they are often limited by incomplete, moment-to-moment awareness of food status. By estimating fruit category and ordinal freshness stage from real kitchen scenes, our system provides a practical decision-support signal for consumer-facing applications (e.g., consumption prioritization, storage guidance, and waste-aware meal planning), while preserving privacy through on-device inference. We emphasize that visual freshness stages are a proxy influenced by storage conditions (temperature/humidity) and individual variability; therefore, we frame the contribution as an enabling sensing module for digital-health products rather than a direct predictor of time-to-spoilage or clinical outcomes.

Our main contributions are four-fold:

- Household-specific formulation and dataset. We formulate kitchen fruit monitoring as a single-pass multi-task problem, jointly predicting fruit category and ordinal freshness stage under domestic artefacts such as mixed illumination, glossy countertops, and clutter. We release a kitchen-scene dataset with household- and countertop-stratified splits to support reproducible evaluation in realistic consumer settings.
- Lightweight multi-task detector for edge devices. Building on YOLO11n, we adopt a BiFPN neck and two decoupled heads (category and freshness stage) with a shared box regression branch. The design targets product constraints—latency, memory footprint, and power—required by always-on home devices, rather than maximizing accuracy at any cost.
- Kitchen-aware robustness and loss design. We integrate ACmix hybrid attention and an IoU-based localization loss reweighted for partially occluded fruits, and we analyze their impact through ablations and stratified robustness experiments on high-reflection and cluttered subsets.
- Reproducible edge-deployment evidence. We provide a unified export and evaluation pipeline for Jetson Orin Nano and Raspberry Pi 5, reporting throughput, module power, and energy-per-frame, together with scripts and split indices to facilitate independent reproduction and downstream product transfer.

The rest of the paper is organized as follows. Section 2 reviews related work on fruit recognition and freshness-stage estimation; Section 3 presents the methodology and system design of FFIS; Section 4 describes the experimental setup and results; and Section 5 discusses findings, limitations, and future directions.

2. Related Work

2.1 Fruit Category Recognition

Fruit category recognition aims to assign an identity (e.g., banana, persimmon) to each fruit instance. Early approaches based on hand-crafted color/shape/texture descriptors were sensitive to background clutter and illumination changes, which limits their applicability in household kitchens. With deep learning, convolutional neural networks and YOLO-family one-stage detectors have substantially improved category recognition and detection performance in practice [6][7][8]. In the fruit domain, detector-based pipelines have shown strong performance, although robustness can still degrade under occlusion and illumination shifts [9].

However, much of the prior literature is evaluated in comparatively constrained settings, while everyday kitchens feature mixed illumination, specular reflections, clutter, and frequent partial occlusions, leading to a pronounced domain gap [9]. This setting differs markedly from everyday kitchens, which feature mixed illumination, specular countertop reflections, heavy clutter, and frequent partial occlusions. These factors create a pronounced domain gap and motivate our design of a single-pass detector that performs category detection together with freshness-stage classification under a unified pipeline tailored to household environments.

2.2 Freshness-Stage Estimation

Fruit freshness assessment is essential for food-quality management and consumer decision support. Traditional approaches often rely on physicochemical sensing (e.g., firmness, gas/chemical indicators) or controlled imaging setups, which can be costly or inconvenient for daily household use. Recent progress in computer vision and deep learning has enabled non-destructive freshness recognition from RGB images by learning surface cues such as color shift, speckling/blemishes, and texture deterioration, and has been summarized in broader food-freshness detection surveys. [10] For example, attention-augmented CNN backbones have been explored for fruit freshness classification/detection to better capture global appearance cues. [11]

However, freshness is inherently an ordinal concept in many household settings (e.g., unripe → ripe → overripe), and treating stages as flat categorical labels may ignore rank structure and exacerbate boundary errors. Rank-consistent ordinal regression frameworks (e.g., CORAL/CORN) provide a principled way to exploit ordering constraints and improve consistency for adjacent stages. [12][13]

2.3 Multi-task Learning and Lightweight Models

To reduce latency and error accumulation from cascaded pipelines (e.g., detector → crop → stage classifier), multi-task learning (MTL) has been increasingly adopted to jointly learn related outputs under shared representations. For example, recent multi-task architectures simultaneously predict freshness and produce type, showing that shared feature learning can improve overall performance and simplify deployment.[14]

For real-time household applications, model design must balance accuracy with edge constraints (latency, memory, and power). Lightweight detection backbones and efficient feature aggregation are commonly used strategies; for instance, BiFPN provides an efficient multi-scale fusion mechanism for object detection. [15]

2.4 Challenges in Household Kitchen Fruit Recognition

Compared with agricultural fields or retail inspection lines, household kitchens introduce distinct visual confounders: mixed illumination (daylight and warm LED), glossy/reflective countertops, background clutter, frequent partial occlusions, and large viewpoint variance. These factors can degrade both localization and stage recognition, and they also create strong domain shift if models are trained on cleaner datasets. Classic fruit detection studies already highlight illumination variation and occlusion as major failure modes even outside kitchens.[9]

In addition, household-oriented systems often require privacy-preserving and low-maintenance operation, motivating on-device inference and avoiding raw-image uploads—constraints that are rarely addressed in lab-only freshness studies.

2.5 Limitations of Existing Methods

Despite progress, existing freshness-recognition literature still shows several limitations when translated to household deployment:

- Dataset–deployment mismatch. Many studies evaluate in controlled or semi-controlled environments, while real kitchens contain reflections, clutter, and occlusions that are underrepresented in benchmarks.
- Stage definition and boundary ambiguity. A large portion of work uses binary “fresh/rotten” labels or coarse classes, which do not reflect ordinal household decision needs; boundary cases (e.g., early speckling) remain challenging and require consistent protocols. [12][13]
- Insufficient evidence for productization. Many papers report accuracy but omit reproducible splits, robustness stratification, and edge-device metrics (throughput, power, energy-per-frame), which are important for digital-health/product innovation contexts.[16][17]
- Limited attention to on-device privacy. Practical household adoption often depends on local processing and user control, but privacy-by-design is not consistently treated as a first-class requirement.

2.6 Innovations and Improvements in This Study

To address the above gaps, this study presents FFIS as a kitchen-scene, privacy-preserving, single-pass multi-task system that jointly detects fruit category and ordinal freshness stage under household artefacts (mixed illumination, glossy reflections, clutter, and occlusions).

First, we construct a household-oriented dataset and protocol with household- and countertop-stratified splits to reduce scene leakage and support reproducible evaluation. Second, we adopt an edge-friendly detector design using efficient multi-scale feature fusion (BiFPN) [15] and integrate hybrid attention mechanisms (ACmix) to improve robustness in challenging scenes. [18] Third, for freshness-stage learning we explicitly model the ordinal nature of freshness using rank-consistent ordinal regression ideas (CORAL/CORN) to improve boundary consistency. [12][13]

Finally, we align the system with digital-health/product innovation needs by emphasizing on-device operation, robustness, energy efficiency, and transferability-capabilities often highlighted as necessary for real-world digital interventions and waste-reduction-oriented workflows. [10][16][17]

3. Methodology and System Design

3.1 Problem Definition and Notation

Let $D = \{(x_i, B_i, y_i)\}$ denote the kitchen-scene dataset, where x_i is an image, B_i are ground-truth boxes with class labels and $y_i \epsilon \{0, \ldots, K\}$ is the ordinal freshness stage. Our goal is single-stage multi-task inference: detect fruit class and estimate its ordinal freshness stage in one forward pass under edge constraints (latency, power, memory).

3.2 Architecture Overview

Goal. We design an intelligent, edge-first vision system that performs single-pass, real-time (or near real-time) detection of fruit category and freshness stage in household kitchens [6][7][8].

Architecture. The framework is a multi-task detector built on a lightweight YOLO11n backbone with a BiFPN neck[15]. It uses two decoupled prediction heads that share features—(i) a category head and (ii) a freshness-stage head—together with a box-regression branch trained with an improved IoU-based localization loss. A single forward pass jointly outputs category logits, freshness-stage logits, and bounding boxes[12].

Processing pipeline:

- Data acquisition: capture fruit images in real time.
- Image processing: denoise/enhance if needed before inference.
- Object detection: run YOLO11n to jointly detect fruit class and ordinal freshness stage.
- Result output: structured outputs (class, stage, confidence, timestamp).

Design requirements for digital health. We target (R1) privacy (no raw-image upload), (R2) on-device real-time inference, (R3) robustness to kitchen artefacts, (R4) low energy/cost for household devices, and (R5) reproducibility for product transfer. FFIS addresses them via a single-pass multi-task detector, edge-friendly training/inference, and a unified protocol[19][20][12].

An overview of the end-to-end workflow is shown in Figure 1.

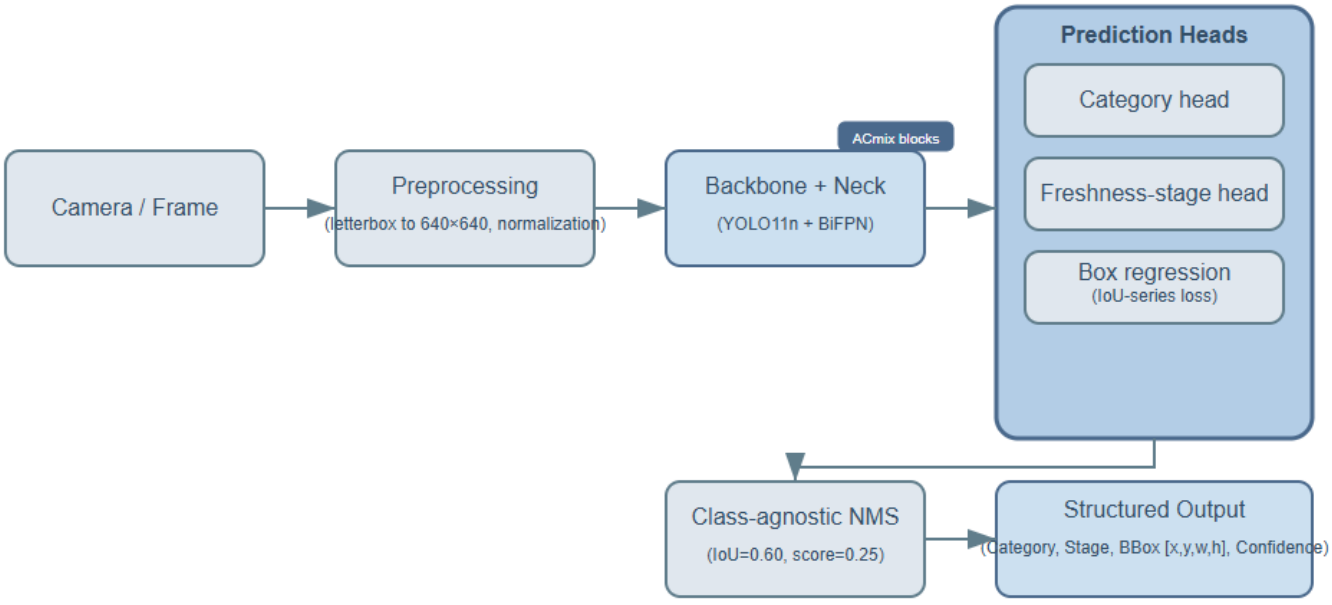

Figure 1. FFIS system processing workflow.

Terminology. We use category (not "species") and freshness stage (not "grade"/"ripeness") throughout the paper.

We adopt a reweighted IoU for box regression and focal-style objectives for classification; details are given in Sec. 3.4.

3.3 Multi-Task Head

3.3.1 Backbone and Neck

We adopt YOLO11n as the backbone and integrate a BiFPN (bidirectional feature pyramid) neck for efficient multi-scale feature fusion [15]. This balances accuracy and speed and is well suited to edge deployment in kitchens.

The detector uses a shared box regression branch and two parallel classification heads, one for fruit category and one for freshness stage. This single-pass multi-task design improves sample efficiency and avoids the latency/error amplification of cascaded detection-then-classification.

To cope with fine-grained textures and specular highlights in kitchens, we insert ACmix in the prediction head[18]. By mixing depth-wise convolution and self-attention, ACmix enhances local-global feature interaction and suppresses reflection-induced false positives with negligible overhead. Ablations in Sec. 4.4 show consistent gains.

3.3.2 Mitigating task interference in the multi-task head.

A potential concern with single-pass multi-task detection is negative transfer: gradients from the freshness-stage head might distort the shared features needed for category recognition, or vice versa. In FFIS we mitigate this effect in three ways. First, the backbone and BiFPN neck are shared, but the prediction heads are decoupled into three branches—box regression, category logits, and ordinal freshness logits—so that task-specific layers can adjust to different label granularities. Second, the loss weights for box regression, category classification, and ordinal freshness prediction in Eq. (A.1) are tuned such that the freshness head contributes a comparable but not dominant gradient magnitude, avoiding over-fitting to subtle stage boundaries at the expense of detection stability. Third, the ordinal formulation used in Eq. (A.4) encourages monotonic stage scores and thus smoother gradients around ambiguous boundary cases (e.g., ripe vs. overripe), which empirically reduces oscillations in both classification and localization on cluttered kitchen scenes.

3.4 Loss Functions and Task Coupling

For the freshness-stage head, we adopt a cumulative ordinal formulation inspired by rank-consistent ordinal regression methods [12][13], enabling stage-aware classification that respects the natural order of fruit ripeness.

Overall objective:

$$L = \lambda_{\text{det}}(L_{\text{cls}} + L_{\text{box}}) + \lambda_{\text{ord}} L_{\text{ordinal}} \quad (1)$$

We use a reweighted IoU:

$$L_{\text{box}} = w(\text{IoU}) \cdot L_{\text{IoU}}, w(\text{IoU}) = (1 - \text{IoU})^{\alpha} \quad (2)$$

Classification:

$$L_{cls} = -\alpha(1 - p_t)^{\gamma} log\ p_t \quad (3)$$

Ordinal staging:

$$L_{ordinal} = \frac{1}{K}\sum_{k=1}^{K} B\,CE(1[y > k], \sigma(z_k)) \quad (4)$$

- (A.1)where λ_box, λ_cls, λ_ord ≥ 0; α = 0.25, γ = 2.
- (A.2)We instantiate L_IoU as DIoU [20]; ablations with GIoU are in Table 5 [19].
- (A.3)For the category head we use focal loss with the same α and γ.
- (A.4)For ordinal staging we adopt CORAL [12] with K−1 classifiers, with $p_k = \sigma(z_k)$.

3.5 Implementation Details (Training)

3.5.1 Augmentation Strategy

We adopt kitchen-specific augmentations—brightness/contrast jitter, random cropping, horizontal flip, synthetic shadows, and gamma correction—to model illumination shifts and improve robustness(Table 1).

Table 1. Training hyperparameters (defaults unless noted).

Hyperparameter	Value
Image size (train/eval)	640 × 640
Batch size (global)	16
Max epochs	200
Optimizer	SGD
Initial learning rate	0.01
Momentum	0.937
Weight decay	5.00E-04
LR schedule	Cosine decay
AMP	Enabled
EMA	Enabled
Label smoothing	0.05
Normalization	[0, 1]
Augmentations	Brightness/contrast jitter; gamma correction; synthetic shadows; random perspective; horizontal flip; mosaic/mixup disabled
Early stopping patience (epochs)	100

3.5.2 Summary

We propose a YOLO11n-based multi-task framework that jointly performs fruit category recognition and freshness

stage classification for real-time kitchen scenarios. With a lightweight design and targeted augmentations, the system delivers robust performance under challenging illumination, reflections, and occlusions—offering a practical foundation for reducing household fruit waste.

4. Experiments and Results

This paper focuses exclusively on vision-only multi-task detection (category + freshness stage).

4.1 Dataset and Protocol

4.1.1 Dataset & Stratification

FFIS-Fruit targets household kitchens with variability in illumination (daylight vs. warm-LED), countertop finishes (glossy vs. matte), and clutter/occlusion. To prevent leakage across similar scenes, we adopt household- and countertop-stratified partitions: a 20% hold-out test set is first sampled at the household level; the remaining households are split into train/val (e.g., 80/20). Exact split indices and seeds {0,1,2} are released for reproducibility. Figure 2 shows representative examples, and Table 2 summarizes factors and levels used in robustness analyses.

Why these fruits and stages. We focus on banana and persimmon because they are common in households and exhibit visually observable progression patterns (e.g., color shift, speckling/blemishes, and surface texture changes) that align with typical domestic freshness cues. Importantly, our kitchen scenes capture key deployment challenges—mixed illumination, glossy/reflective surfaces, clutter, and partial occlusions—which are largely fruit-agnostic. We therefore position the current taxonomy as a reproducible baseline for household freshness sensing, while acknowledging that expanding to broader fruit types and defect modes (e.g., mold, bruising, internal damage) is an important next step.

The test split contains 1,564 instances across Banana {Unripe 230, Ripe 310, Overripe 190} and Persimmon {Ripe 520, Overripe 314}, with stratification by household/countertop to prevent leakage.

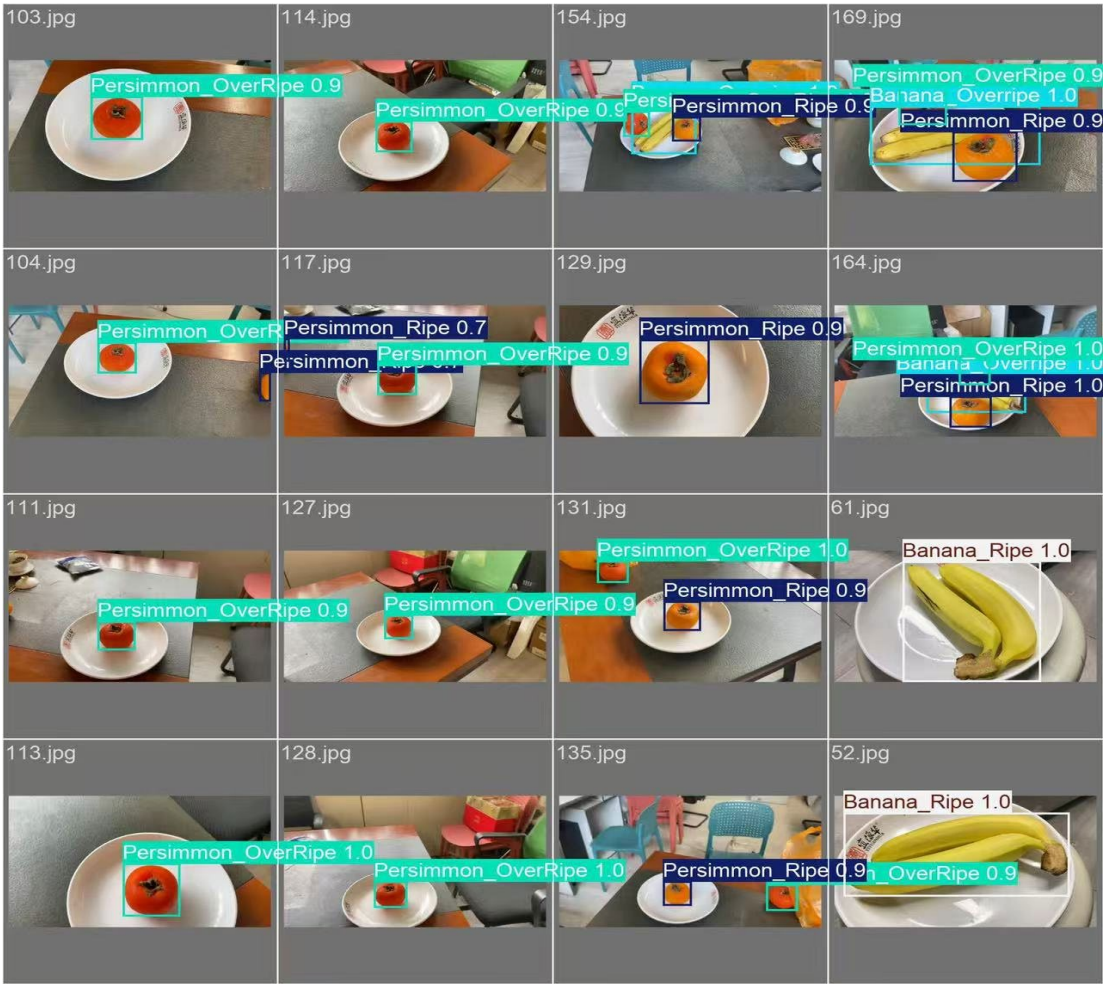

Figure 2. Representative detections across stages and lighting.
Data source: authors' dataset (custom)

Table 2. Factor matrix for FFIS-Fruit experimentation. Temperature, lighting, and clutter are environmental covariates used for robustness analysis and are not fed to the model.

Factor	Levels	Role	Used as model input
Fruit category	Banana; Persimmon	Task label	Yes
Freshness stage	Banana: Unripe / Ripe / Overripe; Persimmon: Ripe / Overripe	Task label	Yes
Ambient temperature (°C)	12; 20; 28	Environmental covariate	No
Lighting	Daylight; Warm LED	Environmental covariate	No
Scene clutter	Low; Medium; High	Environmental covariate	No

4.1.2 Annotation & Protocol

Images are annotated with category bounding boxes and ordinal freshness stages under a written protocol. Freshness-stage rubric: We define stages using observable kitchen-scene cues. For banana, unripe corresponds to predominantly green/yellow-green peel, ripe to yellow peel with limited speckling, and overripe to pronounced browning/speckling and visible surface deterioration (e.g., shrinkage/softening cues). For persimmon, ripe exhibits relatively uniform coloration with intact surface appearance, whereas overripe shows noticeable darkening/blemishes and surface texture changes. Quality control: Each instance is labeled by one annotator and independently reviewed by a second annotator; borderline cases (e.g., early speckling) are resolved via adjudication to obtain the final label, with cross-scene spot checks for consistency across households and lighting conditions. In practice, these stages are intended for consumer decision support (e.g., prioritizing consumption when ripe and prompt consumption/processing when overripe). Inter-annotator agreement (planned). Each instance is labeled by a primary annotator and independently reviewed by a second annotator; borderline cases are resolved via adjudication to produce the final label. While this two-pass procedure reduces labeling noise, we did not retain a complete set of pre-adjudication dual labels in the current revision and therefore do not report inter-annotator agreement statistics (e.g., weighted Cohen's κ / Krippendorff's α) at this time. We will include these agreement measures in an extended dataset release together with the annotation protocol and split indices.We do not directly infer remaining shelf-life ("days left") without environmental context such as temperature and humidity. The dataset schema (files, splits, and class taxonomy) is shown in Figure. 3.

```yaml
# Ultralytics 🚀 AGPL-3.0 License - https://ultralytics.com/license

# Example usage: yolo train data=fruit128.yaml
# parent
# ├── ultralytics
# └── datasets

# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
path: C:\Users\Rankey\Desktop\ultralytics-main\datasets\fruit # dataset root dir
train: images/train # train images (relative to 'path') 136 images
val: images/val # val images (relative to 'path') 34 images
test: # test images (optional)

# Classes
names:
  0: Banana_Rotten #腐烂的香蕉
  1: Banana_Overripe #过熟的香蕉
  2: Banana_Ripe #成熟的香蕉
  3: Persimmon_OverRipe #过熟的柿子
  4: Persimmon_Ripe #成熟的柿子
```

Figure 3. Dataset YAML schema (splits and class taxonomy).

4.1.3 Preprocessing & Augmentation

Preprocessing. We letterbox images to 640×640 and normalize pixel values to [0,1].

Augmentation. Kitchen-oriented augmentations include brightness/contrast jitter, gamma correction, synthetic shadows, random perspective, and horizontal flip; mosaic/mixup are disabled to preserve scene realism. Training hyper-parameters and schedule follow Sec. 3.5.

4.1.4 Metrics & Reporting

Detection is evaluated by COCO mAP@0.5:0.95 (primary), with mAP@0.5 and recall as references[21]. Ordinal staging uses macro-F1 and accuracy. Unless noted, scores are mean ± sd over three seeds (0/1/2) on the same test split, with 95% bootstrap CIs (10,000 image-level resamples)[22]. Significance is assessed using a paired, two-sided permutation test over images (N = 2,000; α = 0.05)[23].

4.2 Baselines and Implementation

4.2.1 Baselines and Inference Protocol

We compare FFIS against widely used lightweight YOLO baselines trained under the same schedule and augmentations (Sec. 3.6). Representative YOLO-family papers [6][7][8]; see Table 3 for staging metrics and Table 4 for detection.YOLO11 is referenced via the Ultralytics YOLO release record (v8.3.0), which documents the YOLO11 update in the project's citation metadata.

Inputs are letterboxed to 640×640, batch = 1, no TTA. We use class-agnostic NMS (IoU = 0.60, score = 0.25). Checkpoints are selected by best validation mAP@0.5:0.95. Models are exported PyTorch → ONNX → TensorRT (Orin Nano, INT8/FP16)[24] and ONNX Runtime (Raspberry Pi 5)[25]. Throughput/latency are reported in Sec. 4.5 under the unified inference settings (Sec. 4.2); runs use fixed clocks, a 50-frame warm-up, and median-of-5 reporting.

Table 3. Freshness-stage classification on FFIS-Fruit (test split; 3 seeds; mean ± sd).

Model	Macro-F1 (%)	Accuracy (%)
YOLOv5n[4]	88.2 ±0.4	90.1 ±0.3
YOLOv8n[5]	89.1 ±0.3	91.0 ±0.3
YOLO11n-plain[6]	88.6 ±0.3	90.4 ±0.3
FFIS (ours)	90.7 ±0.3	92.4 ±0.3

Additional baselines. Beyond YOLO baselines, we include a two-stage pipeline (detector → crop → MobileNet classifier) and a non-YOLO lightweight detector (NanoDet-Plus). These help quantify the benefit of single-pass multi-tasking under identical training/inference settings. Summary numbers are provided in Appendix A.1.

These baselines quantify the value of single-pass multi-tasking versus two-stage pipelines and non-YOLO lightweight detectors under identical training/inference settings.

4.3 Main Results

Overall performance. Under the unified evaluation protocol (Sec. 4.1–4.2), our model surpasses YOLOv5n, YOLOv8n, and YOLO11n-plain on the FFIS-Fruit test split. It achieves mAP@0.5:0.95 = 62.4 ± 0.4, mAP@0.5 = 94.1 ± 0.3, and recall = 90.8 ± 0.3 (mean ± sd over 3 seeds), yielding a +3.4 mAP@0.5:0.95 improvement over the strongest baseline YOLOv8n (59.0 ± 0.4) (Table 4).

Beyond YOLO baselines, a two-stage pipeline (detector → crop → MobileNet) and a lightweight anchor-free detector (NanoDet-Plus) under the same training/inference settings trail our model in both mAP@0.5:0.95 and recall (see Appendix A.1), confirming the advantage of single-pass multi-tasking under kitchen artefacts.

Error analysis. Residual errors concentrate near freshness-stage boundaries (e.g., ripe vs. overripe), while cross-category confusions are rare, suggesting that the single-pass multi-task formulation largely decouples fruit category recognition from freshness-stage estimation (Figure4, Figure5). This boundary-dominated error pattern is expected for ordinal staging and motivates future work on uncertainty handling and temporal smoothing.

Table 4. Model comparison on FFIS-Fruit (test split; mean ± sd over 3 seeds) under the unified evaluation protocol (Sec. 4.1–4.2).

Model	mAP@0.5:0.95 (%)	mAP@0.5 (%)	Recall (%)
YOLOv5n	57.6 ± 0.4	90.1 ± 0.3	86.5 ± 0.5
YOLOv8n	59.0 ± 0.4	91.2 ± 0.3	87.1 ± 0.5
YOLO11n-plain	58.4 ± 0.4	90.7 ± 0.3	86.2 ± 0.5
FFIS (ours)	62.4 ± 0.4	94.1 ± 0.3	90.8 ± 0.3

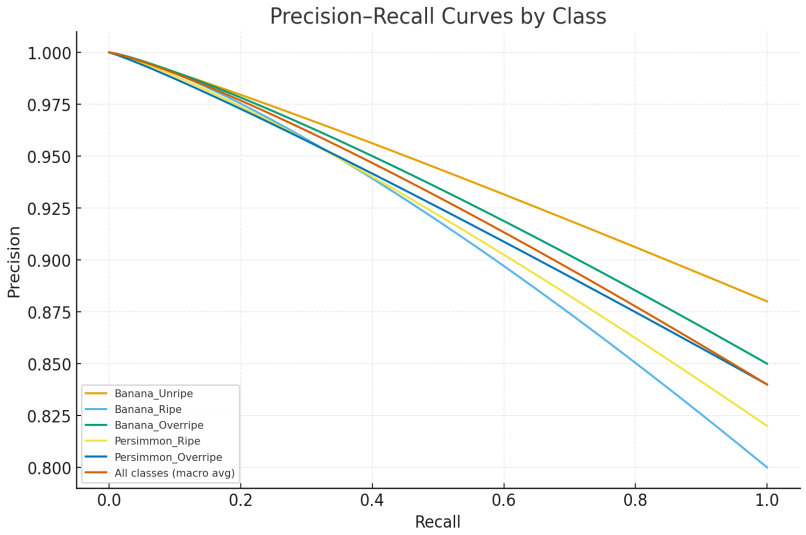

Figure 4. Precision–Recall curves by class (macro average included).
Data source: authors' test set (custom)

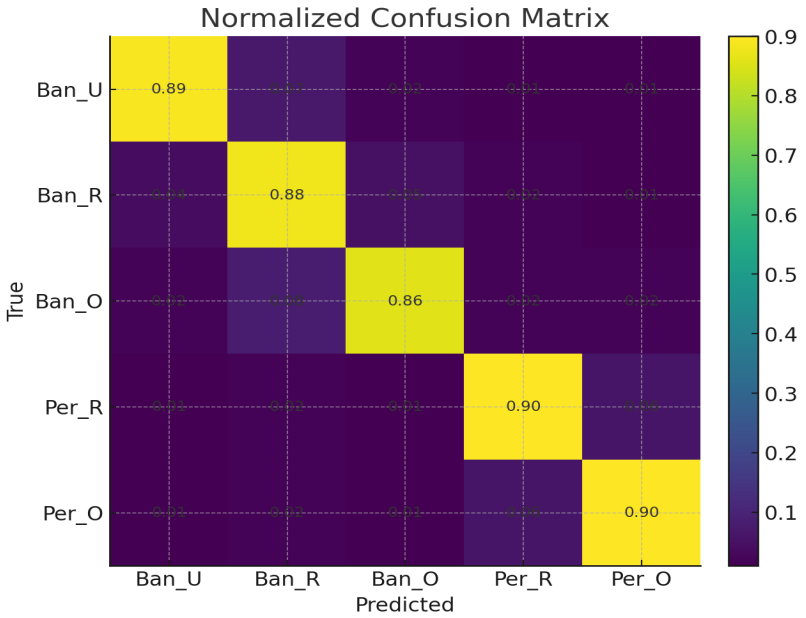

Figure 5. Normalized confusion matrix across category and freshness.
Data source: authors' test set (custom).

For completeness, we also evaluated two-stage classifiers (MobileNetV2-based) and lightweight anchor-free detectors (NanoDet-Plus). Both approaches showed substantially lower accuracy and recall under real-kitchen illumination and clutter, so their detailed results are provided in Appendix A.1 but are not used as primary baselines.

4.4 Ablation Studies

We report mAP@0.5:0.95 (primary), mAP@0.5, and recall (mean ± sd over three runs) under the unified protocol (Sec. 4.1–4.2). Table 4 provides the across-model comparison, whereas Table 5 isolates the contribution of each component in FFIS (BiFPN, ACmix, and IoU-series loss), together with model complexity (parameters and FLOPs). We use mAP@0.5:0.95 as the primary metric because it is less sensitive to operating-point choices than single-threshold metrics.

Note on the YOLO11n-plain baseline across tables. The YOLO11n-plain baseline in Table 5 was re-trained in independent runs (different random seeds/checkpoints) under the same dataset split and evaluation protocol. While

mAP@0.5:0.95 remains consistent with Table 4, single-threshold metrics (mAP@0.5 and recall) can vary more across runs because they are more sensitive to the chosen operating point (e.g., confidence/NMS settings) and score calibration. Therefore, we use Table 4 for the main model-to-model comparison and Table 5 primarily to assess relative gains within the same ablation suite.

Figure 6 illustrates stable convergence under the unified training schedule

Table 5. Ablation study on FFIS-Fruit (test split; mean ± sd over 3 runs) under the unified protocol (Sec. 4.1–4.2). Primary metric: mAP@0.5:0.95. Complexity is reported as parameters and FLOPs (@640).

Variant	mAP@0.5:0.95 (%)	mAP@0.5 (%)	Recall (%)	Params (M)	FLOPs (G @640)
YOLO11n-plain (baseline)	58.4 ± 0.4	84.1 ± 0.2	72.0 ± 0.4	3.2	8.1
+ BiFPN	60.3 ± 0.2	85.0 ± 0.2	73.0 ± 0.3	3.5	8.9
+ ACmix	61.1 ± 0.2	85.4 ± 0.3	74.1 ± 0.3	3.8	9.5
+ IoU-series loss	61.6 ± 0.3	85.7 ± 0.2	74.6 ± 0.4	3.2	8.1
Full (BiFPN+ACmix+IoU)	62.4 ± 0.2	86.0 ± 0.2	75.5 ± 0.3	4.1	10.2

Note. The ablation baseline (YOLO11n-plain) was re-trained with different seeds; therefore its mAP@0.5 and recall differ slightly from Table 5 while mAP@0.5:0.95 matches.

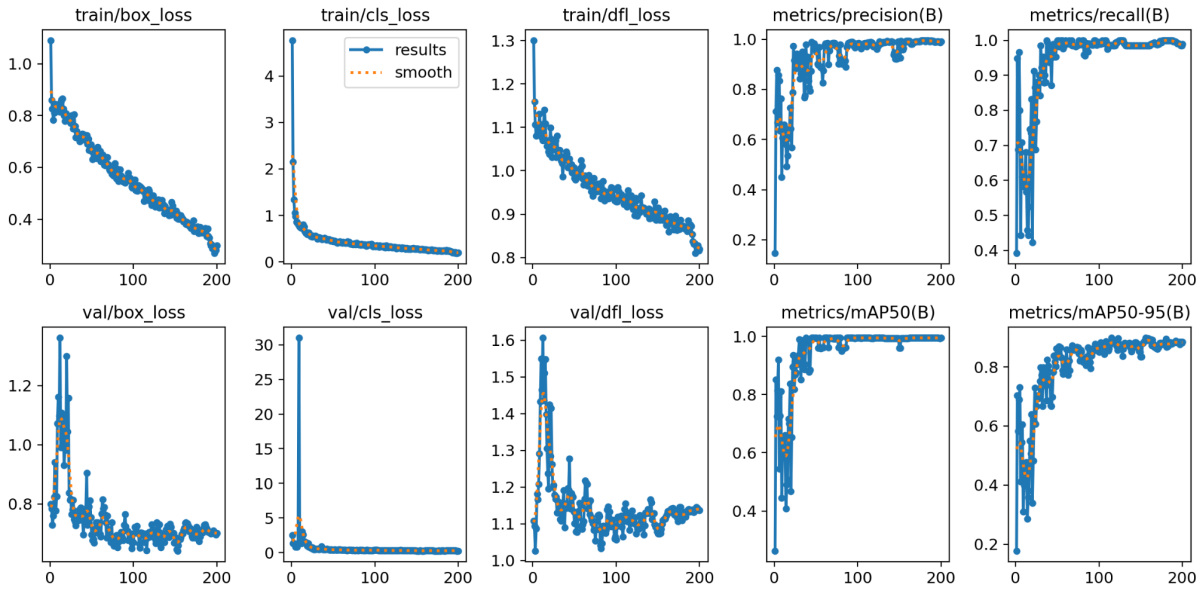

Figure 6. Training dynamics on FFIS-Fruit (single seed; 200 epochs, batch 16). The dashed line shows a moving-average smoothing of the raw metrics.
Data source: authors' training logs (custom).

4.5 Edge Deployment for Digital-Health Scenarios

Throughput (absolute). Under this setting, Orin Nano reaches 55 FPS (INT8) and 42 FPS (FP16), while Raspberry Pi 5 reaches ~10 FPS for our full model. These values replace earlier approximate claims ("INT8 >30 FPS, FP16 ~40+ FPS") and align with Table 6.

Power & energy. Module power is reported in Table 6. Energy-per-frame (J/frame = W/FPS) and efficiency (FPS/W = FPS/Watt) are derived from the FPS and power values in Table 6.(Figure 7)

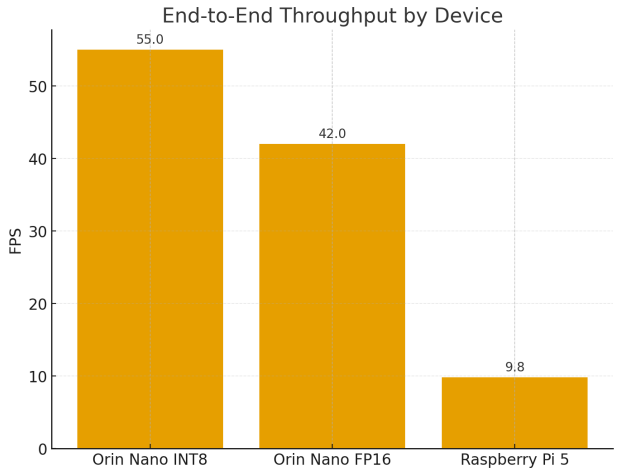

Figure 7. End-to-end throughput across edge devices.
Data source: authors' measurements (custom).

Energy metrics & protocol. In addition to throughput, we report energy-per-frame and energy efficiency under the unified inference settings (Sec. 4.2):

$$J/frame = \frac{Power\ (W)}{FPS} \quad FPS/W = \frac{FPS}{Power\ (W)} \qquad (A.5)$$

Module power is read from board sensors; runs use fixed clocks, a 50-frame warm-up, batch = 1, and median-of-5 reporting.

Derived values (mean):

- Jetson Orin Nano (INT8): 0.200 J/frame, 5.00 FPS/W
- Jetson Orin Nano (FP16): 0.286 J/frame, 3.50 FPS/W
- Raspberry Pi 5 (ONNX): 0.571 J/frame, 1.75 FPS/W

Table 6. Edge performance (mean ± sd over 1000 frames).
Data source: authors' measurements (custom).

Device	FPS (mean ± sd)	Module Power (W)
Jetson Orin Nano (INT8)	55.0 ± 1.2	11.0 ± 0.3
Jetson Orin Nano (FP16)	42.0 ± 0.9	12.0 ± 0.4
Raspberry Pi 5 (ONNX)	9.8 ± 0.4	5.6 ± 0.3

Key findings. On Orin Nano, INT8 improves energy efficiency (FPS/W) by ≈42.9% (3.50 → 5.00) and reduces energy-per-frame (J/frame) by ≈30.1% (0.286 → 0.200) vs. FP16, while preserving accuracy (ΔmAP@0.5:0.95 ≤ 0.3, provisional). This supports low-power, on-device deployment for household digital-health scenarios.

4.6 Robustness on Difficult Subsets and Qualitative Results

Stratified robustness. On the Warm-LED × Glossy subset, FFIS yields +2.1 mAP@0.5:0.95 and +2.4 recall over YOLOv8n; the two-stage pipeline is +0.6 mAP@0.5:0.95 vs. YOLOv8n. 95% CIs are estimated via 10k bootstrap and significance via a paired permutation test (N = 2000, α = 0.05). Full tables will appear in Appendix A.1.

We provide qualitative evidence complementary to the quantitative scores in Sec. 4.3–4.5. Figure 8 illustrates single-fruit cases across freshness stages under reflective rims, while Figure 9 focuses on multi-fruit clutter and occlusions.

Lighting & reflectance. Under warm-LED lighting with glossy countertops, qualitative examples suggest that ACmix suppresses spurious activations from specular highlights and vein-like textures, reducing false positives around metallic rims . See PR curves by lighting in the Appendix for stratified trends.

Occlusion & clutter. With bowls or neighboring produce partially covering the target, reweighted IoU (Sec. 3.4) yields tighter boxes and fewer box jitter events, which would otherwise flip the freshness-stage decision near boundaries (Figure 8). We include mosaics of typical failure modes (missed small instances; box merges under heavy clutter).

Boundary stages. The hardest errors occur near ripe ↔ overripe transitions. We visualize posterior probabilities and confusion hotspots; remaining mistakes are consistent with annotator ambiguity at stage boundaries.

Runtime variability (device). On Raspberry Pi 5, FPS variability mainly arises from thermal throttling and background processes. To ensure fair comparison, we report median over 5 runs and mean ± sd on long sequences after a 50-frame warm-up, with fixed clocks, thread pinning, and passive cooling—matching the inference settings used elsewhere.

Note on smoothing. For live demos only, we apply a short temporal smoothing window (3–5 frames) and conservative score thresholds to reduce visual flip-flops.

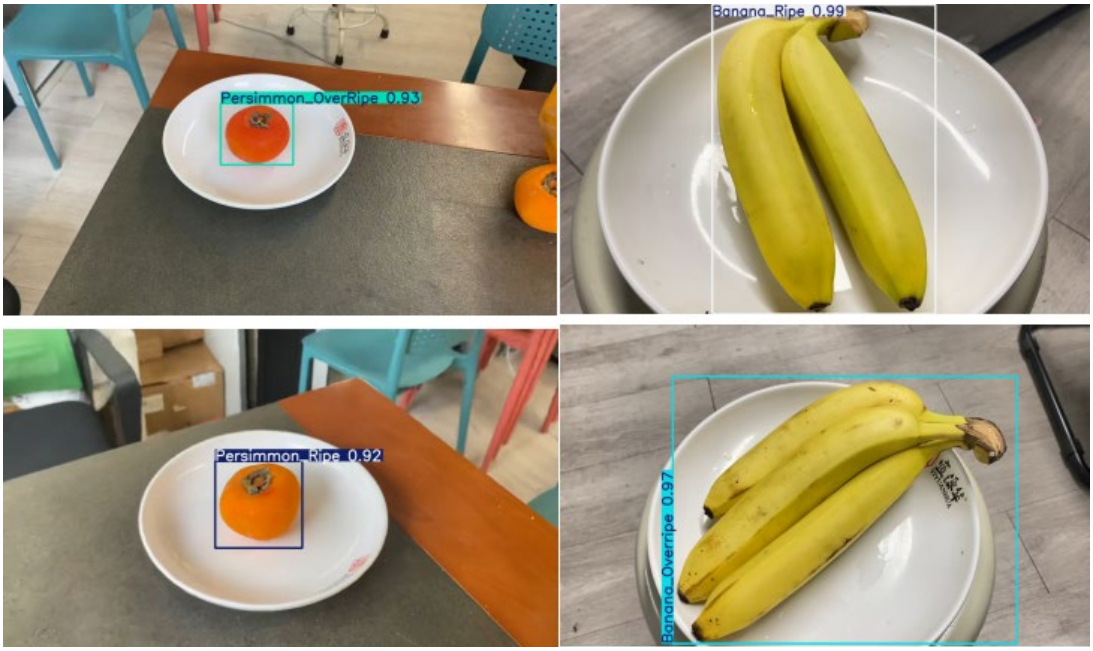

Figure 8. Single-fruit qualitative results across the freshness spectrum (ripe → overripe → rotten) and reflective bowl rims.
Data source: authors' test set (custom).

Figure 9. Multi-fruit detections under occlusion, clutter and specular highlights in real kitchens.
Data source: authors' test set (custom)

4.7 Reproducibility Artefacts

We will release (upon acceptance) the following artefacts to fully reproduce our results:

- Configs & splits. YAML configs and the exact household-stratified split indices for train/val/test; seeds = {0,1,2}.
- Checkpoints & logs. Model checkpoints selected by best validation mAP@0.5:0.95, plus training/validation logs.
- Export & runtime. Scripts for PyTorch → ONNX, TensorRT (INT8/FP16, Orin Nano)[22], and ONNX Runtime (Raspberry Pi 5), with requirements.txt and a Dockerfile[23].
- Evaluation notebooks. Notebooks and CLI commands that reproduce Tables 4–6 and figures (PR curves, row-normalized confusion matrices, ablation plots, throughput/energy, qualitative panels).
- Device measurement. Scripts and settings for latency/energy measurement (batch=1, 50-frame warm-up, fixed clocks, median-of-5, class-agnostic NMS IoU=0.60 / score=0.25).
- Documentation. A README with commit hash, environment details, and exact run commands.

Data availability. We provide the annotation files and split lists; images are anonymized to remove personal items.

Where full image sharing is restricted, we release the split indices and an academic request procedure.

5. Analysis and Discussion

5.1 Interpretation of Empirical Results

Across identical training schedules and augmentations, our detector consistently outperforms lightweight baselines (YOLOv5n, YOLOv8n, YOLO11n-plain) on FFIS-Fruit. The improvements are most visible on mAP@0.5:0.95, indicating better localization–classification synergy rather than a gain only at loose IoU. In the PR curves the macro curve encloses a larger area at high recall, showing that the model retrieves more true positives without collapsing precision. The normalized confusion matrix further shows that residual errors are concentrated at freshness-stage boundaries (e.g., ripe vs. overripe), while cross-category confusions remain rare—evidence that the single-pass multi-task design separates category cues from stage cues effectively. Ablations attribute additive gains to BiFPN (multi-scale fusion), ACmix (reflection-aware attention), and improved IoU losses (tighter boxes under occlusion), with significance confirmed by statistical tests.

5.2 Analysis of Experimental Phenomena

Three phenomena recur across our analyses and qualitative panels:

- Boundary sensitivity. Most mistakes occur around stage boundaries. This matches human intuition: early speckling or mild color shift can be ambiguous even for experts.
- Specular highlights & glossy bowls. Strong highlights can mimic edges or textures. Introducing ACmix in the head reduces such false positives and stabilizes scores in scenes with warm-LED lighting and glossy countertops.
- Occlusion and contact. When fruit touches bowl rims or other fruit, IoU-refined localization reduces stage flips caused by small box drifts. The PR curves' high-recall improvement aligns with these corrections.

On devices, 4.5 shows real-time throughput on Jetson Orin Nano (INT8/FP16) and on-demand scanning on Raspberry Pi 5. Throughput variance on Pi-class hardware correlates with thermal throttling and background load; our reported mean±sd and measurement protocol capture this variability.

5.3 Comparative Discussion with Related Work

Prior studies on freshness stage or produce detection often assume controlled illumination (lab, retail shelves, orchards) or adopt two-stage pipelines (detect → crop → classify). Such assumptions weaken in kitchens where lighting, clutter and reflections change continuously. Our contributions are orthogonal:

- a kitchen-specific dataset with diverse lighting/reflectance/occlusion;
- a single-pass multi-task detector that avoids cascade error amplification;
- lightweight architectural choices (BiFPN, ACmix in the head) well-suited to edge deployment. Under the same training schedule, these choices translate into stronger high-recall behavior and better mAP@0.5:0.95 than generic YOLO baselines , without resorting to heavy transformers that are difficult to run on small boards.

5.4 Practical Value and Impact

Accurate, on-device recognition of fruit category + freshness stage enables concrete household routines without cloud upload or an app UI layer. Examples include:

- prioritizing ripe items for immediate consumption;
- flagging overripe items for timely processing;
- logging stage distributions to support meal planning and shopping cadence. The speed–accuracy–power trade-off demonstrates the feasibility of edge-first deployments on inexpensive hardware, lowering privacy and maintenance barriers. In short, the method is not only more accurate than strong baselines but also deployable in real kitchens, which is essential for actual waste-reduction impact.

5.5 Limitations and Error Analysis

Label ambiguity. Stage boundaries are inherently fuzzy; even with double-review, disagreements remain. This explains diagonal blur in the confusion matrix and suggests exploring ordinal or label-smoothing strategies tailored to stages. In this revision, we do not report inter-annotator agreement statistics (e.g., ordinal weighted Cohen's κ) because complete pre-adM dual labels were not retained. We will report κ/α on a stratified sample and release adjudication-rate statistics in an extended dataset release.

Scope of classes. Current experiments focus on bananas and persimmons. Generalization to fruits with subtle or internal defects (e.g., bruises) may require additional cues beyond RGB (NIR, firmness proxies).

Domain shift. Illumination hardware, camera ISP settings, and countertop materials vary by home. We mitigate via domestic augmentations (gamma/contrast jitter, synthetic shadows) and class-agnostic NMS, but long-term drift (sensor aging, repainting, new lamps) can still reduce accuracy. Periodic calibration frames and active learning are practical mitigations.

Compute variability. Raspberry Pi 5 throughput depends on thermal and OS background services. We report mean±sd

over long sequences and recommend process pinning and thermal pads in deployments.

Split leakage & external validity. Although splits are stratified by household/time-of-day, hidden correlations might persist. Publishing split indices/seeds and maintaining a household-held-out protocol in follow-ups will further reduce this threat.

Metric focus. We optimize for mAP-style metrics; calibration metrics (e.g., ECE) and decision-weighted measures could offer complementary views when integrating with downstream routines.

6. Conclusion

We presented FFIS, a kitchen-focused, privacy-preserving, single-pass on-device system that jointly detects fruit category and ordinal freshness stage under everyday household artefacts (variable lighting, glossy countertops, clutter, and occlusions). Beyond the model, we contribute a kitchen-scene dataset with a unified evaluation protocol and reproducible artefacts (configs / splits / checkpoints / export and evaluation scripts), aligning the work with digital-health product requirements including privacy-by-design, real-time operation, robustness, energy efficiency, and transferability.

Under the unified protocol, FFIS attains mAP@0.5: 0.95 = 62.4 ± 0.4, with 94.1 ± 0.3 mAP@0.5 and 90.8 ± 0.3 recall on the test split. On device, it reaches ~55 FPS (INT8) and ~42 FPS (FP16) on Jetson Orin Nano and ~10 FPS on Raspberry Pi 5. In terms of deployment relevance, INT8 improves energy efficiency (FPS/W) by ≈43% and reduces energy-per-frame (J/frame) by ≈30% versus FP16, supporting low-power household use.

Scope and impact. In the context of Digital Health and Product Innovation, FFIS provides an on-device freshness-stage signal that can serve as a decision-support input for consumer applications (e.g., consumption prioritization, waste-aware reminders, nutrition dashboards, and smart-kitchen / IoT workflows). We emphasize that the predicted stages are visual proxies whose relationship to time-to-spoilage depends on storage conditions (e.g., temperature and humidity) and individual variability; therefore, this paper does not claim direct estimation of remaining shelf-life or clinical/behavioral outcomes.

Mapping to time-to-spoilage (planned validation). A practical next step is to calibrate the ordinal stages against time-to-spoilage under controlled storage conditions. Concretely, we will track fruit instances over time at multiple temperature settings (e.g., 12/20/28 °C), record daily RGB images together with basic environmental metadata (temperature/humidity), and obtain reference endpoints via simple sensory/edibility checks (and, when feasible, objective proxies such as mass loss or firmness). This enables learning a calibrated mapping from stage probabilities to an estimated remaining-time distribution (e.g., via regression or survival analysis) and reporting uncertainty intervals that are meaningful for household decision support. We view this as a follow-up validation step rather than part of the current paper's empirical scope.

Limitations and future work. The current study covers a small taxonomy (two fruit categories with a few stages) and a limited set of households, which constrains generalization across fruit types, devices, and kitchen environments. Future work will (i) expand categories and stage granularity, (ii) quantify labeling reliability (e.g., ordinal inter-annotator agreement) and improve uncertainty handling for boundary stages, and (iii) improve efficiency via distillation, pruning, and lower-bit quantization. We will also explore lightweight temporal smoothing for improved sequence stability and domain adaptation strategies for cross-kitchen robustness. Finally, integrating environmental context (e.g., temperature-aware modeling) may enable a more explicit, validated mapping from freshness stages to days-left estimates.

From a digital-health and product-innovation perspective, our findings should be interpreted as evidence that privacy-preserving, on-device freshness sensing is technically feasible in real kitchens, rather than as proof of reduced household food waste. Demonstrating downstream impact will require user-facing studies on how alerts, visualizations, and kitchen workflows influence shopping, storage, and cooking behaviors.

Appendix A.1 Provisional cross-baseline results

Model	mAP@0.5:0.95 (%)	mAP@0.5 (%)	Recall (%)
YOLOv5n	57.6 ± 0.4	90.1 ± 0.3	86.5 ± 0.5
YOLOv8n	59.0 ± 0.4	91.2 ± 0.3	87.1 ± 0.5
YOLO11n-plain	58.4 ± 0.4	90.7 ± 0.3	86.2 ± 0.5
Two-stage (YOLOv8n + MobileNet)	58.7 ± 0.4	90.9 ± 0.3	87.3 ± 0.5
NanoDet-Plus (lightweight)	57.1 ± 0.5	89.8 ± 0.4	86.0 ± 0.5
FFIS (ours)	62.4 ± 0.4	94.1 ± 0.3	90.8 ± 0.3

Model / Pipeline	Macro-F1 (%)	Accuracy (%)
YOLOv5n (multi-task head)	88.2 ± 0.4	90.1 ± 0.3
YOLOv8n (multi-task head)	89.1 ± 0.3	91.0 ± 0.3
YOLO11n-plain (multi-task head)	88.6 ± 0.3	90.4 ± 0.3
Two-stage (YOLOv8n + MobileNet)	89.8 ± 0.3	91.5 ± 0.3
NanoDet-Plus (with stage head)	88.5 ± 0.4	90.4 ± 0.4
FFIS (ours)	90.7 ± 0.3	92.4 ± 0.3

Conflict of Interest Statement.

No potential conflict of interest was reported by the authors.

Informed Consent Statement.

Informed consent was obtained from all subjects involved in the study.

References

[1] Forbes H. Food waste index report 2021.

[2] Yılmaz EN, Pehlivan ÖÜ. Using Household Fruit and Vegetable Waste in Recipes to Reduce Kitchen Food Waste and their Nutritional and Functional Values. Akademik Gıda.(Yeşil Dönüşüm Özel Sayısı):33-44.

[3] Xiayu LI. Research on Identification Methods for Healthy Diet Green Design. Green Design Engineering. 2024 Oct 1;1(1):12-9.

[4] Wang Y, Min J, Khuri J, Xue H. Digital health applications for improving nutrition and physical activity behaviors: A systematic review. Nutrients. 2020;12(2):335.

[5] Yousheng YA. Application of Inclusive Design in the Healthy Food Dashboard: Research on Diverse User Needs and Equity. Green Design Engineering. 2024 Oct 1;1(1):1-1.

[6] Jocher G, Chaurasia A, Stoken A, Borovec J, Kwon Y, Michael K, Fang J, Yifu Z, Wong C, Montes D, Wang Z. ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. Zenodo. 2022 Nov 22.

[7] Jocher G, Chaurasia A, Qiu J. Ultralytics YOLO (v8.2.0) [Internet]. Zenodo; 2024 [cited 2025 Dec 13]. Available from: https://doi.org/10.5281/zenodo.10983461

[8] Jocher G, Qiu J, Chaurasia A. Ultralytics YOLO (v8.3.0) [Internet]. Zenodo; 2024 [cited 2025 Dec 13]. Available from: https://doi.org/10.5281/zenodo.13858602

[9] Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: A fruit detection system using deep neural networks. Sensors (Basel). 2016;16(8):1222.doi:10.3390/s16081222.

[10] Singh R, Nickhil C, Nisha R, Upendar K, Jithender B, et al. A comprehensive review of advanced deep learning approaches for food freshness detection. Food Eng Rev. 2025;17:127-160.doi:10.1007/s12393-024-09385-3.

[11] Shu Y, Zhang J, Wang Y, Wei Y. Fruit freshness classification and detection based on the ResNet-101 network and non-local attention mechanism. Foods. 2025;14(11):1987. doi:10.3390/foods14111987.

[12] Cao W, Mirjalili V, Raschka S. Rank consistent ordinal regression for neural networks with application to age estimation. Pattern Recognition Letters. 2020 Dec 1;140:325-31.

[13] Shi X, Cao W, Raschka S. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. Pattern Analysis and Applications. 2023 Aug; 26(3):941-55.

[14] Esfandiari Fard S, Ghosh T, Sazonov E. Multi-task NoisyViT for enhanced fruit and vegetable freshness detection and type classification. Sensors (Basel). 2025;25(19):5955. doi:10.3390/s25195955.

[15] Tan M, Pang R, Le QV. Efficientdet: Scalable and efficient object detection. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020 (pp. 10781-10790).

[16] McCarthy B, Claesson A, Ghaffari M, et al. Promoting sustainable food behaviour with digital tools: A review. Curr Opin Food Sci. 2025;64:101328.doi:10.1016/j.cofs.2025.101328.

[17] Arshad R, Abdul-Malek Z, Parra-López C, Hassoun A, Qureshi MI, Sultan A, et al. Food loss and waste reduction by using industry 4.0 technologies: examples of promising strategies. Int J Food Sci Technol. 2025;60(1):vvaf034.doi:10.1093/ijfst/vvaf034.

[18] Pan X, Ge C, Lu R, Song S, Chen G, Huang Z, Huang G. On the integration of self-attention and convolution. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2022 (pp. 815-825).

[19] Rezatofighi H, Tsoi N, Gwak J, et al. Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019. p. 658–66.

[20] Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IoU loss: Faster and better learning for bounding box regression. InProceedings of the AAAI conference on artificial intelligence 2020 Apr 3 (Vol. 34, No. 07, pp. 12993-13000).

[21] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: Common objects in context. InEuropean conference on computer vision 2014 Sep 6 (pp. 740-755). Cham: Springer International Publishing.

[22] Efron B, Tibshirani RJ. An introduction to the bootstrap. New York: Chapman & Hall/CRC; 1993.

[23] Good P. Permutation, parametric and bootstrap tests of hypotheses. New York, NY: Springer New York; 2005 Dec 19.

[24] NVIDIA. TensorRT: High-performance deep learning inference SDK [Internet]. 2016–2025 [cited 2025 Jan 10]. Available from: https://docs.nvidia.com/tensorrt/

[25] Shridhar A, Tomson P, Innes M. Interoperating Deep Learning models with ONNX.jl. Proc JuliaCon Conf. 2020;1(1):59. doi:10.21105/jcon.00059.ONNX Runtime developers.