Multi-Stage Residual Refinement for Apple Segmentation on MinneApple

Keonvin Park
kbpark16@snu.ac.kr
Interdisciplinary Program in Artificial Intelligence,
Seoul National University,
Seoul, South Korea

Jin Hong Mok*
jhmok1024@dgu.edu
Department of Food Science and Biotechnology,
Dongguk University-Seoul,
Goyang, South Korea

Abstract

Accurate and efficient apple segmentation is critical for automated agricultural analysis and yield estimation. The MinneApple dataset has emerged as a standard benchmark for evaluating detection and segmentation performance in orchard environments. While modern convolutional architectures achieve competitive mean Intersection-over-Union (mIoU), the impact of progressive refinement depth on segmentation accuracy and computational efficiency remains underexplored. In this work, we investigate multi-stage residual refinement for apple segmentation on the MinneApple dataset. Starting from a strong UNet-ResNet baseline (1-stage), we introduce a progressive residual correction framework in which each additional stage predicts a residual mask to refine the previous output. Importantly, we increase refinement depth without modifying the backbone, enabling controlled analysis of refinement behavior. Through a systematic study from one to five refinement stages under identical training settings, we observe consistent accuracy gains from 0.6482 mIoU (1-stage) to 0.6904 (3-stage), and up to 0.6971 with five stages. However, improvements are not strictly monotonic, with intermediate saturation observed at four stages. Notably, inference speed remains largely stable across stages (approximately 235–273 FPS in our implementation), indicating that lightweight residual refinement can improve segmentation accuracy without substantial runtime degradation. These results demonstrate that progressive residual correction effectively enhances apple segmentation on MinneApple, while revealing diminishing returns beyond moderate refinement depth. Our findings provide practical guidance for designing efficient multi-stage segmentation systems in agricultural vision applications.

Keywords

Apple segmentation,Multi-stage residual refinement,MinneApple dataset,Agricultural computer vision,Accuracy-efficiency trade-off

ACM Reference Format:
Keonvin Park and Jin Hong Mok. 2026. Multi-Stage Residual Refinement for Apple Segmentation on MinneApple. In *The 3rd InterAI Workshop: "Interactive AI for Human-Centered Robotics at ACM CHI 2026, April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Corresponding author.

The 3rd InterAI Workshop at CHI 2026, Barcelona, Spain

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Accurate fruit segmentation is a fundamental component of automated agricultural analysis, enabling yield estimation, counting, and quality monitoring. In orchard environments, apples often appear densely clustered, partially occluded, and subject to illumination variations, making precise pixel-level localization challenging. The MinneApple dataset [5] has emerged as a widely used benchmark for evaluating apple detection and segmentation methods under realistic orchard conditions.

Recent advances in deep convolutional neural networks have significantly improved object detection and segmentation performance in agricultural settings [6, 11]. Modern architectures, often built upon residual backbones [4] pretrained on large-scale datasets such as ImageNet [2], achieve strong frame-wise accuracy on MinneApple. However, most prior works focus on backbone improvements or detection frameworks, while comparatively less attention has been paid to understanding how progressive refinement depth affects segmentation quality under controlled architectural settings.

Multi-stage refinement has been widely adopted in computer vision to iteratively improve predictions, particularly in dense and cluttered scenes. Rather than redesigning increasingly complex backbones, refinement-based approaches aim to correct residual errors from previous predictions. Recent work in infrastructure inspection demonstrated that a two-stage residual learning framework (U-Net + Pix2Pix) significantly improves crack boundary precision while maintaining real-time performance on robotic platforms [9]. Specifically, residual correction improved mIoU from a baseline segmentation network to 73.9% while preserving deployability in dynamic environments [9]. This study highlights the effectiveness of residual-based refinement in handling thin, low-contrast structures and boundary ambiguities.

Despite these advances, a systematic investigation of refinement depth remains underexplored in agricultural segmentation benchmarks such as MinneApple. While prior studies validate the effectiveness of a single refinement stage, it remains unclear how segmentation accuracy evolves as refinement depth increases beyond two stages, particularly under heavy occlusion and small-instance regimes typical of orchard scenes.

In this work, we revisit apple segmentation on MinneApple from the perspective of progressive residual correction. Starting from a strong UNet-ResNet baseline, we introduce a lightweight multi-stage residual refinement framework in which each additional stage predicts a correction mask that refines the previous output.

Importantly, we increase the number of refinement stages without altering the backbone architecture, enabling controlled analysis of how segmentation accuracy evolves with refinement depth.

Through a systematic empirical study from one to five refinement stages, we observe consistent mIoU improvements from 0.6482 (1-stage) to 0.6971 (5-stage), with performance gains saturating beyond moderate depth. Our analysis reveals that residual refinement primarily improves boundary localization and small-instance recovery, while excessive stages yield diminishing returns relative to computational complexity.

These findings provide practical guidance for designing efficient segmentation systems for agricultural applications. By isolating the effect of refinement depth on MinneApple, our study clarifies how progressive residual correction contributes to segmentation performance without introducing heavy architectural modifications.

2 Related Work

2.1 Fruit Detection and Segmentation in Agriculture

Deep learning has significantly advanced fruit detection and segmentation in agricultural environments. Early work such as DeepFruits [11] applied convolutional neural networks for fruit localization in orchard settings. Subsequent studies have explored improved detection architectures and large-scale agricultural surveys [6], highlighting the importance of robust fruit perception under varying illumination and occlusion.

The MinneApple dataset [5] established a benchmark for apple detection and instance segmentation in orchard scenes, enabling systematic evaluation of modern deep architectures. While detection accuracy has steadily improved, achieving precise pixel-level segmentation in densely clustered scenes remains challenging due to occlusions, overlapping fruits, and scale variation.

2.2 Deep Semantic and Instance Segmentation

General-purpose segmentation architectures such as Fully Convolutional Networks (FCN) [7], U-Net [10], and DeepLab [1] have demonstrated strong performance across diverse segmentation benchmarks. Residual backbones [4], pretrained on large-scale datasets such as ImageNet [2], are commonly adopted to enhance feature representation.

More recent methods, including Mask R-CNN [3], enable instance-level segmentation and have been applied to agricultural datasets. However, improvements are often achieved through increasingly complex backbone architectures, transformer-based encoders, or feature pyramid enhancements, rather than controlled studies of refinement mechanisms.

2.3 Multi-Stage and Residual Refinement

Multi-stage refinement has been widely used to iteratively improve dense predictions. Approaches such as stacked hourglass networks [8] and iterative refinement modules aim to progressively correct prediction errors. In segmentation, residual refinement strategies predict correction masks that refine coarse outputs, improving boundary localization and small-object recovery.

Recent work in infrastructure inspection demonstrated that residual learning can be effectively deployed in real-time robotic systems. A two-stage U-Net + Pix2Pix framework was shown to enhance crack boundary precision while maintaining practical inference speed [9]. By learning residual correction maps rather than full segmentations, the model improved thin-structure localization and reduced false positives in challenging environments. These findings highlight the practical value of residual refinement in dense, cluttered, and low-contrast visual scenarios.

Despite their demonstrated effectiveness in both general vision tasks and applied inspection systems, systematic evaluation of refinement depth remains limited. In particular, while prior studies validate one or two refinement stages, the relationship between refinement depth, segmentation accuracy, and computational efficiency has not been thoroughly analyzed in agricultural benchmarks such as MinneApple.

2.4 Efficiency and Practical Deployment

In agricultural applications, segmentation models must balance accuracy and computational efficiency for real-time deployment in field robotics and yield estimation systems. Larger backbones and transformer-based models improve accuracy but often incur substantial computational overhead. Similarly, multi-stage refinement introduces additional processing cost, raising practical concerns for embedded or mobile systems.

Understanding how lightweight residual refinement strategies influence the accuracy–efficiency trade-off is therefore critical for real-world deployment. Our work differs from prior studies by isolating refinement depth as the primary experimental variable. Rather than introducing heavier backbones or additional supervisory signals, we progressively increase residual refinement stages on a fixed UNet-ResNet backbone, enabling controlled analysis of accuracy–efficiency trade-offs on the MinneApple benchmark.

3 Methods

3.1 Overview

Figure 1 illustrates the overall architecture of the proposed multi-stage residual refinement framework. The model first produces an initial segmentation prediction using a UNet-style ResNet-18 backbone (Stage-1). Subsequent stages iteratively refine the prediction by learning residual corrections conditioned on both the original image and the previous stage output.

3.2 Problem Formulation

Given an input RGB image $X \in \mathbb{R}^{3\times H\times W}$ from the MinneApple dataset, our goal is to predict a binary segmentation mask $Y \in \{0,1\}^{H\times W}$ indicating apple pixels.

Let $\hat{Y}$ denote the predicted mask. Model parameters are optimized by minimizing a pixel-wise segmentation loss between $\hat{Y}$ and the ground-truth mask Y. Performance is evaluated using mean Intersection-over-Union (mIoU).

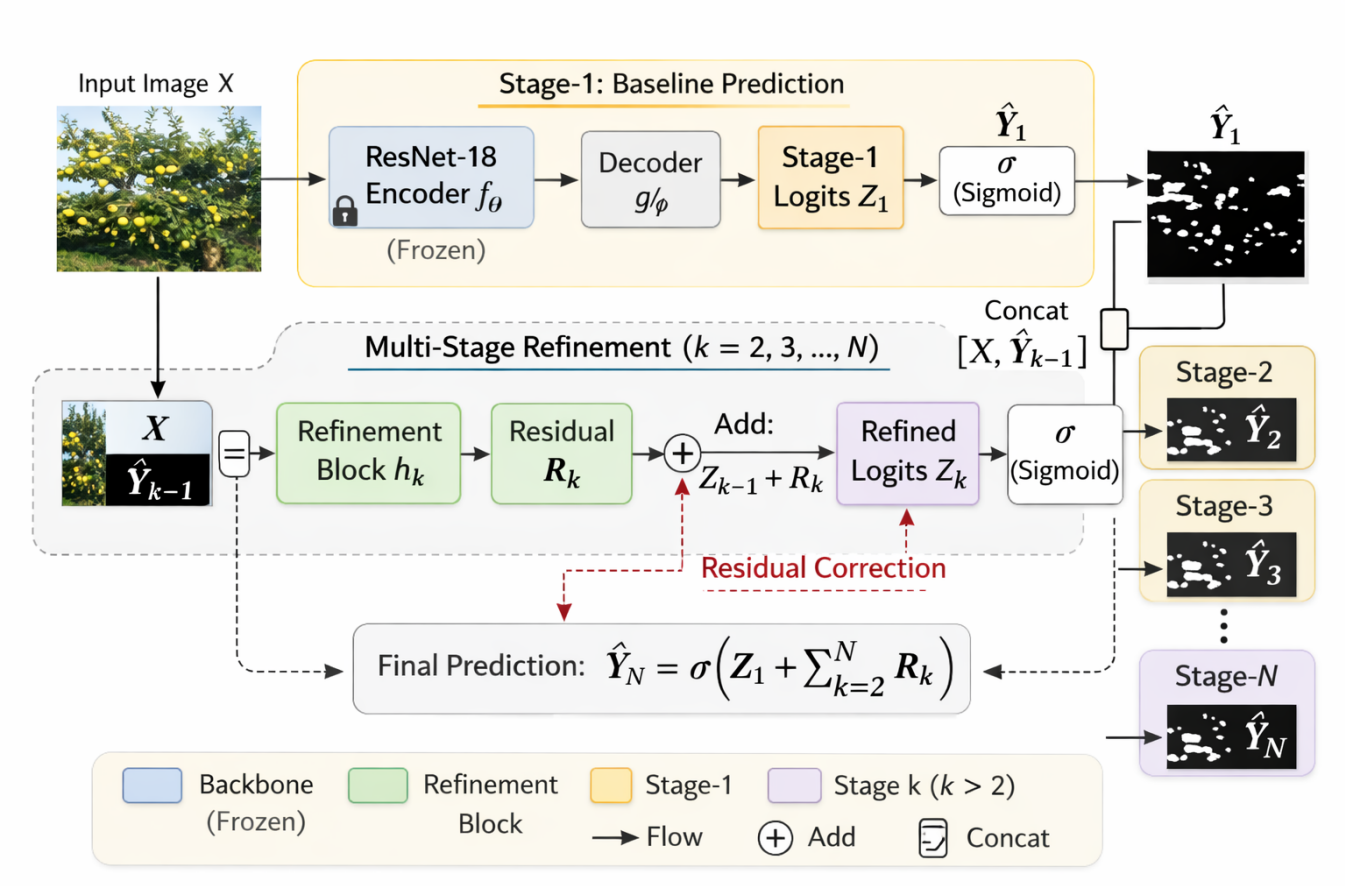

Figure 1: Overall architecture of the proposed multi-stage residual refinement framework. Stage-1 generates an initial segmentation mask using a UNet-ResNet backbone. Subsequent refinement stages iteratively predict residual corrections based on the original image and previous-stage prediction, producing progressively refined masks.

3.3 Stage-1 Baseline: UNet-ResNet

The first stage (Stage-1) follows a standard UNet-style architecture with a ResNet-18 encoder backbone [4], initialized with ImageNet pretraining [2].

The encoder extracts hierarchical feature representations:

$$F = f_\theta(X),$$

where f_θ denotes the ResNet encoder.

A lightweight decoder upsamples feature maps to the original resolution and produces pixel-wise logits:

$$Z_1 = g_\phi(F),$$

where $Z_1 \in \mathbb{R}^{1\times H\times W}$.

The corresponding probability map is obtained via sigmoid activation:

$$\hat{Y}_1 = \sigma(Z_1).$$

Stage-1 is trained using a combination of binary cross-entropy and Dice loss.

3.4 Multi-Stage Residual Refinement

As shown in Figure 1, subsequent stages progressively refine the segmentation output through residual correction.

For stage $k \geq 2$, a refinement module takes as input the original image X concatenated with the previous prediction $\hat{Y}_{k-1}$:

$$R_k = h_k([X, \hat{Y}_{k-1}]),$$

where $h_k(\cdot)$ is a lightweight convolutional refinement block and R_k represents the predicted residual.

The refined logits are computed as:

$$Z_k = Z_{k-1} + R_k,$$

and the updated prediction becomes:

$$\hat{Y}_k = \sigma(Z_k).$$

Rather than re-estimating the segmentation mask from scratch, each stage focuses on correcting residual errors from the previous stage.

3.5 Multi-Stage Architecture

For an N-stage model, the final prediction is:

$$\hat{Y}_N = \sigma\left(Z_1 + \sum_{k=2}^{N} R_k\right).$$

All refinement blocks share the same architectural structure but use independent parameters. Importantly, the Stage-1 backbone remains frozen during multi-stage training. Only refinement blocks are optimized when $N > 1$.

This design isolates the effect of progressive residual correction and allows controlled analysis of refinement depth without increasing encoder capacity.

3.6 Efficiency Measurement

To evaluate deployment feasibility, we measure inference speed in frames per second (FPS) under identical hardware conditions. This enables direct comparison of the accuracy–efficiency trade-off across different refinement depths.

4 Data

4.1 MinneApple Dataset

All experiments are conducted using the MinneApple dataset [5], a publicly available benchmark for apple detection and segmentation in orchard environments. The dataset contains high-resolution RGB images captured under natural outdoor conditions, exhibiting dense fruit clustering, partial occlusion, varying illumination, and significant scale diversity. These characteristics make MinneApple a challenging benchmark for robust segmentation.

Pixel-level segmentation masks are provided only for the training portion of the dataset. The official test set contains RGB images without publicly released ground-truth annotations.

4.2 Training Protocol

Segmentation masks are available only for the training portion of the MinneApple dataset. Therefore, model training is performed directly on the annotated training set using paired RGB images and pixel-level masks.

Instance-level annotations are merged into a single foreground class (apple), and the task is formulated as binary semantic segmentation (apple vs. background). All quantitative performance metrics (mIoU) reported in this work are computed on the annotated training data.

This setup reflects a practical training scenario where only labeled training images are available, while the official test set is used solely for qualitative evaluation.

4.3 Test Inference

Because ground-truth masks are not available for the official test set, quantitative evaluation cannot be performed on those images. Therefore, test-set results are presented qualitatively to demonstrate visual generalization and segmentation behavior in unseen scenes. Quantitative metrics reported in this paper are computed solely on the annotated training data.

5 Experiments

5.1 Implementation Details

All models are implemented in PyTorch. We use a ResNet18-based UNet as the Stage-1 backbone, initialized with ImageNet-pretrained weights. For multi-stage experiments, the Stage-1 network is frozen and only the residual refinement blocks are trained.

Training is performed using the Adam optimizer with a learning rate of 1×10^{-4}. The loss function is binary cross-entropy with logits. All images are resized to a fixed spatial resolution for consistent training and inference.

To ensure reproducibility, all experiments are conducted with a fixed random seed. Performance is evaluated using mean Intersection-over-Union (mIoU), computed via a confusion-matrix-based aggregation over all pixels.

5.2 Evaluation Metrics

Since the segmentation task is formulated as binary (apple vs. background), mIoU is computed as:

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

In addition to segmentation accuracy, we measure inference speed in frames per second (FPS) to analyze the computational overhead introduced by additional refinement stages.

5.3 Multi-Stage Residual Refinement Study

We investigate the effect of residual refinement depth by varying the number of stages from 1 to 5.

- **1-Stage**: Baseline UNet (no refinement)
- **2-Stage**: UNet + 1 residual refinement block
- **3-Stage**: UNet + 2 residual refinement blocks
- **4-Stage**: UNet + 3 residual refinement blocks
- **5-Stage**: UNet + 4 residual refinement blocks

Each refinement block predicts a residual correction map conditioned on the input image and the previous stage's mask logits. The residual is added to the current logits to iteratively improve segmentation quality.

5.4 Quantitative Results

Table 1 reports mIoU and FPS across different stage configurations.

Our results show that introducing residual refinement consistently improves mIoU compared to the single-stage baseline. Performance gains are most noticeable between Stage-1 and Stage-3, while deeper configurations exhibit diminishing returns.

At the same time, FPS decreases as the number of refinement stages increases, reflecting the trade-off between accuracy and computational efficiency.

These findings suggest that moderate refinement depth (e.g., 3-stage) provides a favorable balance between segmentation quality and runtime performance.

5.5 Qualitative Analysis

Figure 2 presents qualitative comparisons across stage configurations. Multi-stage refinement improves boundary sharpness and reduces small false-positive regions in dense apple clusters.

However, overly deep refinement occasionally introduces minor artifacts, indicating that excessive correction depth may lead to overfitting or instability.

Overall, residual refinement enhances mask consistency and boundary precision under challenging orchard conditions.

6 Results

6.1 Quantitative Segmentation Performance

Table 1 reports segmentation accuracy and inference speed across refinement depths.

Introducing residual refinement significantly improves mIoU compared to the single-stage baseline (0.6482). Performance increases to 0.6836 with two stages and peaks at 0.6971 with five stages.

Notably, inference speed remains high across all configurations (236–273 FPS), indicating that lightweight residual refinement introduces minimal computational overhead.

While improvements from Stage-1 to Stage-3 are substantial, gains become marginal beyond three stages, suggesting diminishing returns from excessive refinement depth.

Table 1: Segmentation performance across refinement stages on MinneApple.

Stages	mIoU	FPS
1-Stage	0.6482	236.68
2-Stage	0.6836	235.61
3-Stage	0.6904	237.42
4-Stage	0.6861	250.89
5-Stage	**0.6971**	**273.53**

6.2 Confusion Matrix Analysis

Confusion-matrix-based evaluation reveals that multi-stage refinement primarily reduces false positives in background regions and false negatives along object boundaries. Improvements are most noticeable in dense apple clusters where instance boundaries overlap.

6.3 Qualitative Analysis

Figure 2 presents qualitative comparisons of segmentation outputs across different refinement stages on representative MinneApple samples.

The single-stage baseline (Stage-1) already captures the majority of apple instances, demonstrating that the backbone model provides a strong initial localization prior. However, small-scale inconsistencies are observable in densely clustered regions, where minor boundary irregularities and fragmented mask regions occasionally appear.

As additional residual refinement stages are introduced, mask regions become progressively more coherent. In particular, multi-stage refinement:

- Produces more compact and spatially consistent foreground regions,
- Slightly improves boundary smoothness around individual apple instances,
- Reduces isolated pixel noise in background areas.

The visual differences between Stage-3 and Stage-5 are subtle but consistent, indicating diminishing yet stable improvements as refinement depth increases. Importantly, refinement does not introduce noticeable over-segmentation or structural artifacts, suggesting that residual corrections act conservatively by refining existing predictions rather than drastically altering them.

Overall, qualitative observations align with quantitative results, confirming that residual multi-stage learning improves segmentation coherence and boundary stability under dense orchard conditions without sacrificing structural integrity.

7 Conclusion

In this work, we introduced a multi-stage residual refinement framework for semantic segmentation and evaluated its effectiveness on the MinneApple dataset. By progressively refining intermediate predictions through lightweight residual modules, the proposed approach consistently improves segmentation accuracy while maintaining high inference speed.

Experimental results demonstrate that residual refinement substantially enhances mIoU compared to a single-stage baseline, with performance peaking at five refinement stages. Importantly, the computational overhead remains minimal, preserving real-time inference capability (over 230 FPS across all configurations). These findings indicate that iterative residual correction is an efficient strategy for improving segmentation coherence without sacrificing efficiency.

Residual learning has previously demonstrated effectiveness in other dense-structure segmentation domains, such as infrastructure crack detection, where two-stage refinement significantly improved boundary precision under challenging visual conditions. Building upon these insights, our work systematically analyzes refinement depth in an agricultural benchmark setting. Unlike prior studies that validate one or two refinement stages, we provide controlled empirical evidence on how segmentation performance evolves as refinement depth increases.

Quantitative analysis and qualitative observations reveal that multi-stage refinement reduces false positives in background regions and improves boundary consistency in densely clustered orchard scenes. While gains diminish beyond three refinement stages, improvements remain stable and do not introduce structural artifacts, suggesting that residual learning operates conservatively by correcting systematic errors rather than overfitting noise.

Overall, this study highlights the practical effectiveness of lightweight multi-stage residual learning for dense-instance segmentation in agriculture. By isolating refinement depth as the primary experimental variable, we clarify its contribution to segmentation accuracy–efficiency trade-offs without modifying backbone capacity. Future work may explore adaptive refinement depth, boundary-aware supervision, and integration with instance-level segmentation frameworks to further enhance performance in complex orchard environments.

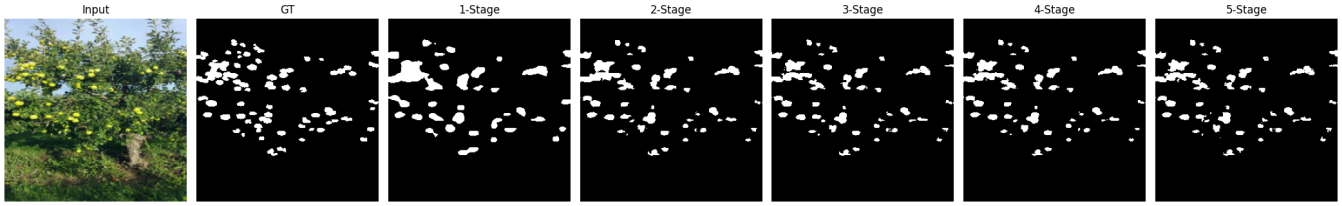

Figure 2: Qualitative comparison of segmentation results across refinement stages on the MinneApple dataset. From left to right: Input image, ground-truth mask (GT), and predictions from Stage-1 to Stage-5 models. Multi-stage residual refinement progressively improves mask coherence and boundary smoothness, particularly in densely clustered apple regions.

References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *ECCV* (2018).

[2] Jia Deng, Wei Dong, Richard Socher, and et al. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask R-CNN. In *ICCV*.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[5] Nicolas Häni, Patrick Roy, and Volkan Isler. 2019. MinneApple: A Benchmark Dataset for Apple Detection and Segmentation. *arXiv preprint arXiv:1909.06441* (2019).

[6] Anish Koirala, K. B. Walsh, Zongyuan Wang, and et al. 2019. Deep Learning for Real-Time Fruit Detection and Quality Assessment: A Review. *Computers and Electronics in Agriculture* 162 (2019), 219–234.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*.

[8] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*.

[9] Emmanuella Ogun, Yong Ann Voeurn, and Doyun Lee. 2026. A Real-Time Mobile Robotic System for Crack Detection in Construction Using Two-Stage Deep Learning. *Sensors* 26, 530 (2026). doi:10.3390/s26020530

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*.

[11] Inkyu Sa, Zongyuan Ge, Feras Dayoub, and et al. 2016. DeepFruits: A Fruit Detection System Using Deep Neural Networks. *Sensors* 16, 8 (2016), 1222.