Lightweight Image Segmentation for Sustainable Smart Farming using Cloud–Edge AI Collaboration

G. Thirumalaiah[1], WeiWei Jiang[2], Shashi Kant Guptha[3]

[1] Lincoln University of College, Malaysia; [2] Beijing University of Posts and Telecommunications, Beijing, China; [3]Lincoln University of College, Malaysia

[1]Annamacharya University,Department of Electronics and Communication Engineering,Rajampet-516115,Andhrapradesh,India

pdf.thirumalaiah@lincoln.edu.my ; jww@bupt.edu.cn, shashigupta@lincoln.edu.my

---

Abstract: Precision agriculture requires accurate, real-time crop and disease segmentation to enable site-specific management with minimal resource use. This paper consolidates our proposed Cloud–Edge collaborative approach for lightweight semantic segmentation on resource-constrained devices. We combine classical image pre-processing (median filtering and resizing) with a compact deep model tailored for edge deployment and evaluate the pipeline on four horticultural crops (mango, sweet orange, chilli, and tomato). We demonstrate feasibility on a Raspberry Pi edge node, present qualitative results, confidence/precision–recall curves, and a normalized confusion matrix extracted from the experimental dashboard. The study highlights the design choices that favor energy- and memory-efficient inference without cloud dependence, and discusses remaining challenges in generalization, dataset diversity, and multispectral sensing.

Keywords: Edge AI, image segmentation, MobileNetV3, precision agriculture, multispectral imaging, Raspberry Pi

---

Introduction

Precision agriculture demands pixel-level understanding of field imagery for tasks such as crop–weed discrimination and disease monitoring[1]. While multispectral/hyperspectral data improve plant separability, many state-of-the-art segmentation models remain heavyweight, rely on cloud GPUs, and are unsuitable for rural deployments with intermittent connectivity[2]. Edge devices (e.g., drones, field robots, Raspberry Pi) impose tight constraints on compute, memory, and energy, motivating compact architectures, pruning/quantization, and task-aware feature engineering. This work consolidates a presentation-driven study into a manuscript, detailing problem scope, related work, a lightweight pipeline, and results from a Raspberry Pi prototype[3].

Related work

We summarize representative crop/weed segmentation efforts emphasizing model compactness and deployability. Table 1 lists studies curated in the presentation[3].

Table 1. Compares this work with the related work or previous research by other researchers

No.	Title (Author, Year)	Techniques Used	Limitations
1	RDS_Unet: Improved U-Net for Crop–Weed Segmentation[4]	U-Net + ResNeXt-50 encoder, deformable convolutions, attention	Heavy model; not deployable on low-power edge devices
2	TinySegformer: Lightweight Visual Segmentation for Pest Detection[5]	Transformer-CNN hybrid, quantized model for Jetson Nano	Complex training; accuracy lower on smaller pests
3	Lightweight Multispectral Crop–Weed Segmentation[6]	CNN-Transformer hybrid; RGB+NIR+RedEdge	Requires expensive sensors; untested on embedded platforms
4	WeedsGalore Dataset for Maize Weed Segmentation[7]	UAV dataset + CNN baselines (U-Net, DeepLabv3)	Region-specific crops; baseline models not edge-optimized
5	YOLOv8 for Crop–Weed Segmentation[8]	YOLOv8 segmentation; high FPS object-based detection	Bounding box only; lacks pixel-level accuracy for weeds
6	MSFCA-Net for Crop–Weed Segmentation[9]	Lightweight CNN with multi-scale strip convolutions and attention	Limited lighting scenarios; requires more generalization
7	U-Net + CRF on Sunflower Weed Dataset[10]	U-Net + Conditional Random Fields (CRF); post-processing	Not tested for edge deployment; dataset not diverse
8	Multispectral Image Synthesis using GANs[11]	Conditional GANs for synthetic RGB+NIR images	Model trained on synthetic data; may overfit or generalize poorly
9	WeedMap: UAV Multispectral Weed Mapping[12]	SegNet with multispectral orthomosaic input	High GPU dependence; no quantization or optimization applied
10	weedNet: Multispectral Dense Classification via SegNet[13]	SegNet on UAV-based multispectral data	Limited FPS; outdated Jetson TX2 edge deployment trial only

3. Materials and Methods

3.1 Data and Task Setup

We considered four crops—mango, sweet orange, chilli, and tomato—using leaf imagery to support segmentation and disease analysis. The presentation indicates use of multispectral/vegetation-index channels (up to 14) to inject task-relevant priors into an encoder–decoder network[14].

3.2 Pre-processing

Images were resized to 224×224 pixels and denoised using a median filter to suppress salt-and-pepper noise while preserving edges[15].

3.3 Model Architecture and Segmentation

A lightweight segmentation pipeline based on a compact CNN with MobileNetV3-style modules was used to balance accuracy and latency on embedded hardware[16]. The network follows an encoder–decoder topology, with task-aware channels (e.g., vegetation indices) concatenated at the input[17].

3.4 Disease Stage Assessment

Predicted disease categories were mapped to interpretable stage descriptors to guide pesticide recommendations. Table 2 summarizes stage features for representative mango diseases extracted from the slides.

Table 2. Disease-stage features (extracted from the presentation)

Name of the Leaf	Disease Name	Stage-1 Features	Stage-2 Features	Stage-3 Features
Mango	Anthracnose	Small, angular, irregular shape brown or black spots. Short holes.	Elongated brown spots(20-25mm diameter).	Dark brown or blacks spots (>5mm in diameter). Severe leaf curling
Mango	Die Back	Small, circular, brown or gray spots (1-3mm in diameter).	Irregular shaped brown or gray spots (5-10mm in diameter). Leaves begin to wilt and curl.	Leaves turn completely brown or grey. Severe leaf curly.
Mango	Phoma Blight	Small, Circular &Dark spots(1-5mm) in diameter. 2-10mm in diameter spots may appear and	Large spots and spots is in 5-20 mm in diameter. Leaves turn yellow and 10-50% of leaves area is affected.	Infected leaves turn dark brown or black and become brittle with leaf thickness of 0.1-0.5 mm.

		cottony white growth available underside of leaves.		fungus on leaf growth

3.5 Edge AI System and Deployment

The inference pipeline was deployed to a Raspberry Pi edge node as part of a Cloud–Edge–Device workflow. A block diagram and device/dashboard photographs are included for context[18], [19].

4. Results

We report qualitative inputs/outputs, confidence/precision–recall diagnostics, and a normalized confusion matrix captured from the evaluation dashboard. Wherever possible, artifacts are inserted directly from the presentation. Quantitative values were not provided beyond the curves.

Figure 1. Mango Leaves

Figure 2. Sweet Orange Leaves

Figure 3. Chilli Leaves

4.1 Input images and purpose.

Figure X shows example leaf images that were used as inputs to our model. The panels group the samples by crop: mango, sweet orange, and chilli. These images were chosen to represent the variety we expect in real conditions—different viewing angles, backgrounds, and lighting. Showing the raw inputs helps the reader see what the model actually receives before any processing or prediction. Each crop has typical visual traits that help the model tell them apart. Mango leaves are usually longer and narrower with a strong central vein. Sweet orange leaves are more oval and often appear slightly broader and glossier. Chilli leaves are generally slim and lance-shaped. These differences in overall shape, margin smoothness, and vein patterns give the model clear cues even when the background or light changes.

Some leaves look fully green and uniform, while others show yellowing, dark spots, or patchy areas. These are common signs of stress or disease. The model can use such color and texture changes to separate healthy tissue from affected regions. In simple terms, smoother, even-colored areas are more likely to be healthy, while sharp color breaks and rough textures often indicate problems that the segmentation step needs to outline.

4.2 Pre-processing and robustness.

Before inference, each image is lightly cleaned and resized to a standard size. A median filter reduces small "salt-and-pepper" noise without blurring edges, and resizing puts all inputs on a common scale. This makes the later steps more stable and helps the model learn features that work across different cameras and distances. The variety in Figure X—angles, lighting, and backgrounds—also encourages the model to generalize and avoid overfitting to a single setting.

By covering both crop differences (mango vs. orange vs. chilli) and condition differences (healthy vs. stressed), the figure explains why the model can achieve reliable segmentation on unseen images. The inputs set the stage for the later results (curves and confusion matrix) by making clear the visual challenges the system must handle.

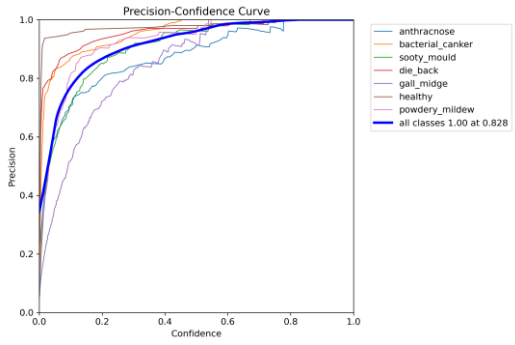

Figure 4 : F1 Confidence Curve

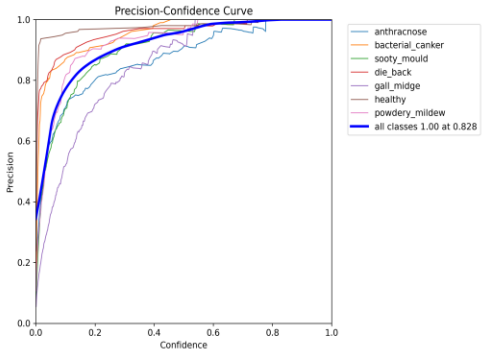

Figure 5 : Precision Confidence Curve

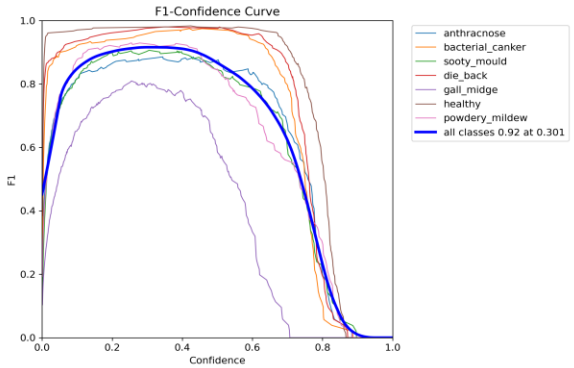

Figure 6: Precision – Recall Curve

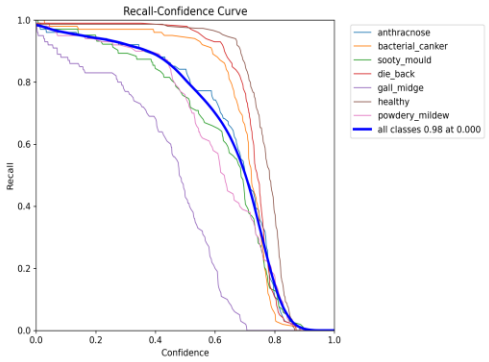

Figure 7: Recall – Confidence Curve

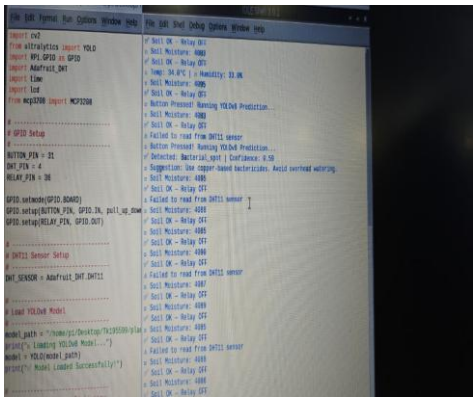

Figure 8 : Dashboard View at Physical node

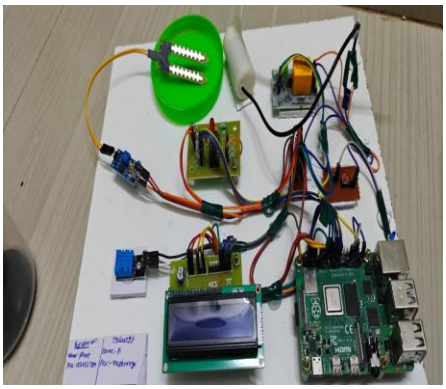

Figure 9 : Physical view at Edge device-

Our model outputs a confidence value for every pixel. By changing the decision threshold from low to high, we can see how the balance between correct and incorrect detections changes. The F1–Confidence curve (Figure 4) shows a clear peak at a mid-range threshold. At very low thresholds the model marks too many pixels as positive, so F1 is lower. As the threshold increases, F1 improves and then drops again when the model becomes too strict. We use the threshold at the peak as the default operating point because it gives a good balance between missing positives and adding false positives.

The Precision–Confidence curve (Figure 5) increases steadily as the threshold rises. This means that stricter decisions reduce false positives and make predictions more precise. The trade-off is visible in the Recall–Confidence curve (Figure 7), which goes down with higher thresholds because some true positive pixels are no longer accepted. Together, these two plots make the trade-off simple to tune: if the application wants fewer false alarms, choose a higher threshold; if the application wants to avoid missed detections, choose a lower threshold.

The Precision–Recall curve (Figure 6) summarizes performance over all thresholds. Points on the upper-right part of this curve are preferred because they give high precision with good recall. In practice, we select an operating point near the “elbow” of the curve or use the threshold from the F1 peak in Figure 4. This gives a single, consistent setting that works well across different images.

4.3 Operational evidence

The dashboard view (Figure 8) shows the evaluation screen used at the physical node. It records the same curves seen in the paper and helps us repeat the tests with the same settings. The edge device photo (Figure 9) confirms that the complete pipeline runs on the Raspberry Pi setup. This demonstrates that the method is practical for field deployment without constant cloud access.

4.4 Recommendation

Unless there is a special cost for false positives or false negatives, we recommend using the F1-optimal threshold from Figure 4 as the default. If the use case needs very clean masks (for example, to avoid unnecessary spraying), move the threshold slightly higher to gain precision. If the use case needs early warnings and broader coverage, move the threshold slightly lower to gain recall.

References

[1] N. Rai *et al.*, "Applications of deep learning in precision weed management: A review," *Comput. Electron. Agric.*, vol. 206, Mar. 2023, doi: 10.1016/j.compag.2023.107698.

[2] H. M. Sahin, T. Miftahushudur, B. Grieve, and H. Yin, "Segmentation of weeds and crops using multispectral imaging and CRF-enhanced U-Net," *Comput. Electron. Agric.*, vol. 211, Aug. 2023, doi: 10.1016/j.compag.2023.107956.

[3] K. Hu *et al.*, "Deep learning techniques for in-crop weed recognition in large-scale grain production systems: a review," *Precis. Agric.*, vol. 25, no. 1, pp. 1–29, Feb. 2024, doi: 10.1007/S11119-023-10073-1.

[4] G. Castellano, P. De Marinis, and G. Vessio, "Weed mapping in multispectral drone imagery using lightweight vision transformers," *Neurocomputing*, vol. 562, Dec. 2023, doi: 10.1016/j.neucom.2023.126914.

[5] A. Iqbal, M. Sharif, M. Yasmin, M. Raza, and S. Aftab, "Generative adversarial networks and its applications in the biomedical image segmentation: a comprehensive survey," *Int. J. Multimed. Inf. Retr.*, vol. 11, no. 3, pp. 333–368, Sep. 2022, doi: 10.1007/S13735-022-00240-X.

[6] A. Makhlouf, M. Maayah, N. Abughanam, and C. Catal, "The use of generative adversarial networks in medical image augmentation," *Neural Comput. Appl.*, vol. 35, no. 34, pp. 24055–24068, Dec. 2023, doi: 10.1007/S00521-023-09100-Z.

[7] H. M. Sahin, T. Miftahushudur, B. Grieve, and H. Yin, "Segmentation of weeds and crops using multispectral imaging and CRF-enhanced U-Net," *Comput. Electron. Agric.*, vol. 211, Aug. 2023, doi: 10.1016/j.compag.2023.107956.

[8] Y. Li *et al.*, "An improved U-net and attention mechanism-based model for sugar beet and weed segmentation," *Front. Plant Sci.*, vol. 15, 2024, doi: 10.3389/FPLS.2024.1449514/FULL.

[9] C. Yun, Y. H. Kim, S. J. Lee, S. J. Im, and K. R. Park, "Artificial intelligence-based semi-supervised crop and weed semantic segmentation," *Appl. Soft Comput.*, vol. 183, Nov. 2025, doi: 10.1016/j.asoc.2025.113662.

[10] J. Lin *et al.*, "FG-UNet: fine-grained feature-guided UNet for segmentation of weeds and crops in UAV images," *Pest Manag. Sci.*, vol. 81, no. 2, pp. 856–866, Feb. 2025, doi: 10.1002/PS.8489.

[11] S. Sonawane and N. N. Patil, "Comparative performance analysis of YOLO object detection algorithms for weed detection in agriculture," *Intelligent Decision Technologies*, vol. 19, no. 1, pp. 507–519, Jan. 2025, doi: 10.3233/IDT-240978.

[12] A. Ehrampoosh *et al.*, "Intelligent weed management using aerial image processing and precision herbicide spraying: An overview," *Crop Protection*, vol. 194, Aug. 2025, doi: 10.1016/j.cropro.2025.107206.

[13] M. Habib, S. Sekhra, A. Tannouche, and Y. Ounejjar, “New segmentation approach for effective weed management in agriculture,” *Smart Agricultural Technology*, vol. 8, Aug. 2024, doi: 10.1016/j.atech.2024.100505.

[14] Y. Zhang and C. Lv, “TinySegformer: A lightweight visual segmentation model for real-time agricultural pest detection,” *Comput. Electron. Agric.*, vol. 218, Mar. 2024, doi: 10.1016/j.compag.2024.108740.

[15] J. Cui, F. Tan, N. Bai, and Y. Fu, “Improving U-net network for semantic segmentation of corns and weeds during corn seedling stage in field,” *Front. Plant Sci.*, vol. 15, 2024, doi: 10.3389/FPLS.2024.1344958/FULL.

[16] M. Shoaib *et al.*, “An advanced deep learning models-based plant disease detection: A review of recent research,” *Front. Plant Sci.*, vol. 14, 2023, doi: 10.3389/FPLS.2023.1158933/FULL.

[17] S. D. Khirade and A. B. Patil, “Plant disease detection using image processing,” *Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015*, pp. 768–771, Jul. 2015, doi: 10.1109/ICCUBEA.2015.153.

[18] T. S. Poornappriya and R. Gopinath, “RICE PLANT DISEASE IDENTIFICATION USING ARTIFICIAL INTELLIGENCE APPROACHES,” *INTERNATIONAL JOURNAL OF ELECTRICAL ENGINEERING AND TECHNOLOGY (IJEET)*, vol. 11, no. 10, pp. 392–402, Dec. 2020, doi: 10.17605/OSF.IO/R5JXA.

[19] W. B. Demilie, “Plant disease detection and classification techniques: a comparative study of the performances,” *J. Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/S40537-023-00863-9.