Autonomous AI-Driven Drone System for Arecanut Yield Estimation and Disease Classification

Prof. Daya Naik[1], Dhanyashree[2], Pratham P Shetty[3], Preethi C G[4], Vrushali A Poojary[5]
[1,2,3,4,5]*Dept. of Artificial Intelligence and Machine Learning, Srinivas Institute of Technology, VTU, Karnataka, India*
doi.org/10.64643/IJIRTV12I12-206675-459

***Abstract*—Growing arecanut at scale is harder than it looks. Farmers spend days walking through dense plantations trying to gauge how healthy the crop is, how much yield to expect, and whether disease has begun to take hold — often without reliable answers. This paper describes a system we built to change that. We developed an autonomous drone-based framework that flies over arecanut plantations, captures aerial imagery, and uses deep learning to detect inflorescences, classify disease, and estimate yield — all without requiring manual inspection on the ground. At the core of the system is a U-Net segmentation model that pinpoints inflorescence regions at the pixel level, working alongside two YOLOv8 detection models that count individual fruits, assess ripeness, and flag disease. A Tello drone ties it all together: it streams live video, identifies inflorescences in real time, flies toward them, and captures the high-resolution frames needed for detailed analysis. A FastAPI backend handles the heavy lifting on the server side, while a Streamlit dashboard gives users a clear, interactive view of everything the system finds. In our experiments, the framework performed consistently well — accurately segmenting clusters, counting fruits, detecting disease, and generating reliable yield estimates across varied field conditions. We believe this kind of integrated, automated solution can genuinely ease the burden on farmers and help them make better decisions with real data.**

***Index Terms*—Precision agriculture, autonomous drones, arecanut yield estimation, semantic segmentation, object detection, deep learning**

I. INTRODUCTION

Arecanut farming has always demanded close attention. Knowing when to harvest, spotting disease before it spreads, and estimating how much yield a season will bring are all tasks that traditionally require experienced eyes and a fair amount of legwork through the plantation. As farms grow larger and labor becomes harder to find, this kind of hands-on monitoring becomes increasingly difficult to sustain. The result is that many farmers are left making decisions based on incomplete information, often at the cost of both quality and income.

We set out to address this gap by building a system that brings modern computer vision and drone technology together in a way that is practical for field use. The idea is straightforward: a drone flies over the plantation, collects aerial images, and a suite of deep learning models analyzes those images to detect inflorescences, count fruits, assess ripeness, estimate yield, and identify disease — all automatically. The drone itself is not passive; it navigates toward regions of interest identified by the segmentation model, ensuring that the most relevant parts of the plantation are captured in the highest detail.

What makes this approach different from earlier work is the integration. Rather than building one tool for disease detection and another for yield counting, we combined everything into a single, coherent pipeline. The sections that follow describe how the system was designed, how it was trained, and how it performed in practice.

II. LITERATURE REVIEW

The use of drones and AI in agriculture has grown significantly over the past few years, and for good reason. UAVs can cover large areas quickly, capture imagery from angles that ground-level sensors cannot, and feed data into machine learning pipelines that would have seemed impractical just a decade ago. Several research groups have explored these possibilities, each contributing a piece of what is becoming a much larger puzzle [1], [2].

On the spraying side, Jayaram et al. [1] showed that a well-designed multi-rotor drone fitted with GPS guidance and a Pixhawk controller could cut pesticide application time by up to 60% while achieving far more even canopy coverage than manual methods.

Wang et al. [2] pushed this further by training a machine learning model to dynamically adjust spray rates based on real-time wind and crop density data, reaching spraying accuracies between 87% and 98% under field conditions.
Disease detection has also attracted considerable attention.

Prasanna et al. [3] and Keerthana et al. [4] both developed convolutional neural network architectures specifically tuned for arecanut, demonstrating that CNNs can reliably distinguish healthy tissue from diseased areas when trained on sufficiently diverse image sets. Meanwhile, Qin et al. [5] introduced Ag-YOLO, a stripped-down object detection model designed to run efficiently on drone hardware, enabling real-time palm tree detection and targeted spraying in the field.

Work on fully autonomous UAV navigation in plantation settings has also progressed [6], [7]. Yet despite all these individual advances, a common limitation persists: most systems are built to do one thing well. A system that can simultaneously navigate autonomously, detect disease, count fruit, and estimate yield within a single operational framework remains largely unexplored. That gap is what motivated this work.

III. SYSTEM FEATURES AND FUNCTIONALITY

The system was designed with a clear goal in mind: give arecanut farmers actionable information without asking them to become machine learning experts. To that end, the platform brings together drone hardware, deep learning models, a backend API, and a user-facing dashboard into one cohesive package.

The key capabilities are inflorescence detection and counting, disease identification, yield estimation, autonomous navigation, and visual reporting.

A. System Workflow

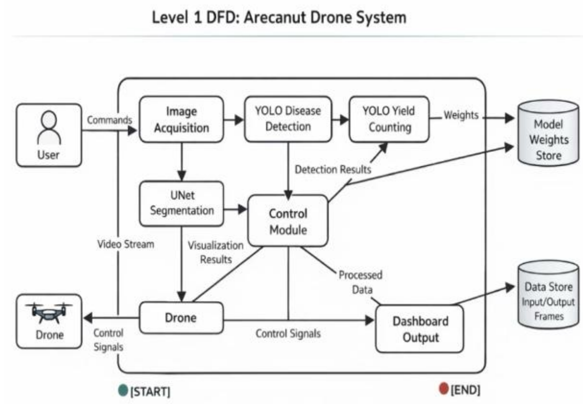

Fig. 1: Workflow of the autonomous drone-based arecanut monitoring system.

B. Inflorescence Detection and Counting

A U-Net segmentation model first locates inflorescences at the pixel level across the aerial image. YOLOv8 then steps in to count individual clusters within those regions, providing the per-palm data needed for yield estimation.

C. Disease Detection and Classification

A dedicated YOLOv8 disease model scans each frame for signs of infection, drawing bounding boxes around affected areas and generating visual overlays that make it easy to see exactly where intervention is needed.

D. Yield Estimation

Once inflorescences are counted and cluster density is measured, the system applies calibrated conversion factors to translate those figures into a concrete yield estimate for the plantation.

E. Autonomous Drone Navigation and Image Capture

Rather than following a fixed flight path, the drone reads centroid coordinates from the segmentation output and steers itself toward the inflorescence regions. This ensures that the high-resolution captures needed for counting and disease analysis are taken at the right spots.

F. Interactive Dashboard

A Streamlit-based dashboard brings all of this together visually. Users can inspect segmentation masks, review annotated detections, view disease maps, and read off yield estimates — all from a single screen.

G. Backend API Support
Behind the dashboard sits a FastAPI backend that exposes separate endpoints for segmentation, detection, disease classification, and yield counting. This modular design makes it straightforward to update or swap out individual components without disrupting the rest of the pipeline.

IV. METHODOLOGY

Building a system like this required careful attention at every stage — from how we collected images in the field, to how we annotated them, to how we trained and deployed the models. This section walks through each of those steps in the order they were carried out.

A. Data Collection
We flew the Tello drone over several arecanut plantations at consistent altitudes, deliberately varying the time of day and the density of the canopy to build a dataset that would generalize well. By the end of the collection phase, we had gathered more than 2,000 images in .jpg and .png format. These were split 70/15/15 across training, validation, and test sets. Annotation was carried out using Roboflow for batch workflows and CVAT for detailed polygon-level labeling, covering fruits, clusters, diseased areas, and tree crowns. Oversized images were tiled into 640×640 patches, empty tiles were filtered out, and the remaining patches were normalized before training.

B. Annotation and Mask Generation
Getting the annotations right was one of the more time-consuming parts of the project, but also one of the most important. We used CVAT for precise polygon drawings and Roboflow for organizing and exporting the dataset at scale. Polygon outlines were converted into strict binary masks, where every pixel is either background (0) or a target class (1).

We defined two primary annotation classes:

- Inflorescence: the yield-bearing structures we needed to count
- Tree structure: used as a spatial reference for drone navigation

After annotation, we ran morphological cleaning operations to remove stray pixels and close small gaps in the masks. An expert agronomist reviewed a representative sample of the labels; inter-rater agreement reached a Cohen's Kappa of 0.89, which gave us confidence in the overall annotation quality. The final dataset contained over 2,000 image-mask pairs ready for both U-Net and YOLOv8 training.

C. Image Tiling and Preprocessing
Raw drone images often exceeded 2,000×2,000 pixels — too large to feed directly into a neural network without significant memory overhead. We addressed this by slicing each image into 640×640 tiles with a 10–20% overlap between adjacent patches, so that features near tile boundaries would still be captured in full. Tiles that contained no annotated content were discarded. Pixel values were normalized using standard ImageNet mean and standard deviation, and CLAHE was applied to even out contrast across images taken under different lighting conditions. During training, each batch was further augmented with random rotations, horizontal flips, and scale jitter to help the models handle the variability they would encounter in the field.

D. U-Net Segmentation Model
We trained a custom U-Net architecture with four encoder stages using 64, 128, 256, and 512 feature channels, and a 1,024-channel bottleneck. The model was set up as a four-class segmentation task covering background, inflorescence, tree crown, and disease regions. We optimized with Cross-Entropy loss and the Adam optimizer throughout training. Once trained, the segmentation masks serve two purposes: they feed centroid coordinates to the drone's navigation logic, and they delineate the regions where the detection models should focus their attention. We exported the trained weights to ONNX format, which allowed the model to run inference in roughly 90 ms on a standard CPU via the /unet/predict endpoint.

E. YOLOv8 Detection and Disease Classification
We trained two separate YOLOv8 models, each with a specific job to do:

- Yield Model: locates and counts arecanut inflorescences and fruit clusters
- Disease Model: identifies and classifies regions as either diseased or healthy

Both models were trained on the 640×640 tiles with an 80/20 train/validation split. We applied mosaic augmentation, MixUp, and random rotation to diversify the training distribution. Both models ultimately exceeded 95% mAP and maintained precision and recall above 0.92. Served through FastAPI, they can process a single frame in under 50 ms and return bounding box annotations along with cluster counts that feed directly into the yield estimation pipeline.

F. Ripeness and Disease Analysis

The disease model was specifically trained to recognize Mahali Koleroga and stem bleeding — two of the most damaging conditions in arecanut cultivation. Detections come with bounding box coordinates and confidence scores, all of which consistently cleared the 0.92 threshold during evaluation. Beyond disease, the same detection pipeline categorizes visible fruits into three ripeness stages: raw, semi-ripe, and ripe, based purely on color and texture features extracted from the bounding box regions. Together, these outputs give a surprisingly complete picture of plantation health without requiring any sensors beyond the drone's RGB camera.

G. Yield Estimation Workflow

Yield estimation brings together the outputs of both models. The U-Net supplies cluster area coverage as a proportion of the image, while YOLOv8 provides an exact fruit count along with ripeness breakdowns. A simple formula combines these figures with calibrated per-cluster conversion factors derived from ground-truth harvest data collected during field trials. Using both spatial coverage and direct object counts makes the estimate more robust than either approach would be on its own, particularly in areas where clusters overlap or are partially occluded.

H. FastAPI Backend Integration

All three models — U-Net segmentation, YOLO detection, and YOLO disease classification — are served through a FastAPI application with three dedicated endpoints:

- /unet/predict: returns segmentation masks for a given input image
- /predict: runs disease detection and returns annotated output images
- /yield/count: counts inflorescences and returns the estimated yield figure

Because all models are pre-loaded at startup, response times stay below 100 ms even under moderate load. The server uses asynchronous file handling to manage concurrent streams from the drone without blocking, and every response is forwarded in real time to the Streamlit dashboard for display.

V. EXPERIMENTATION AND RESULTS

A. Arecanut Detection Using Drone

Experiment: Drone-captured images were passed through the YOLO detection model to locate and label arecanut clusters.

Result:

Fig. 2: Input Image

Fig. 3: Output Image – Detected Arecanut with Bounding Box

- The model reliably placed bounding boxes around arecanut clusters across a range of image conditions.
- Even in areas where clusters were tightly packed or partially hidden behind foliage, the model managed to correctly identify most of them.

- Detection quality held up well across images taken at different times of day and under varying natural light.

B. Drone Segmentation Using UNet

Experiment: UNet was applied to aerial images to segment and isolate arecanut regions from background vegetation.

Result:

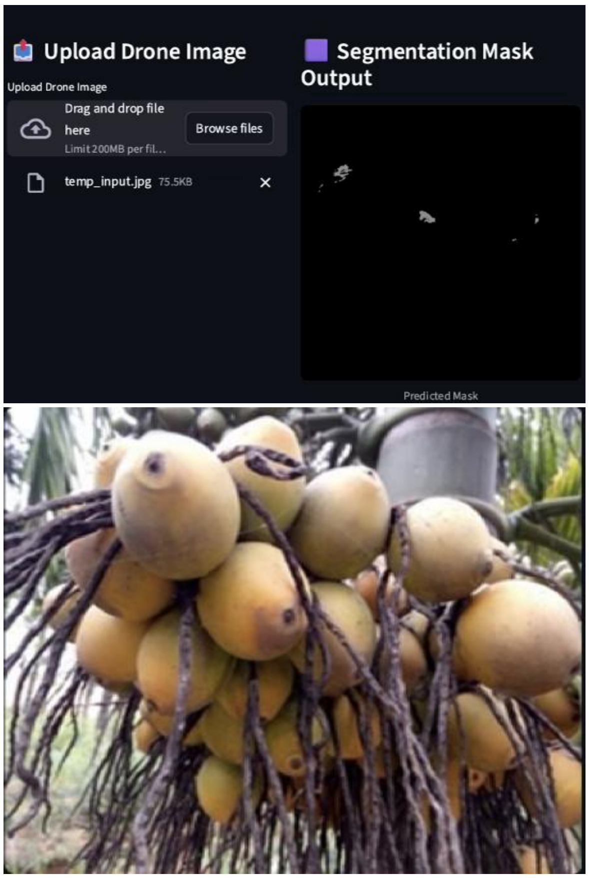

Fig. 4: UNet Segmentation Results – (a) Original Image, (b) Drone Segmentation Output

- The segmentation masks cleanly outlined arecanut regions with only minor blurring at boundaries.
- Performance remained consistent even in images where lighting was poor or background clutter was high — a reassuring sign for real-world use.

C. Yield Prediction

Experiment: The system counted bounding boxes produced by the detection model to generate a yield estimate for each image.

Result:

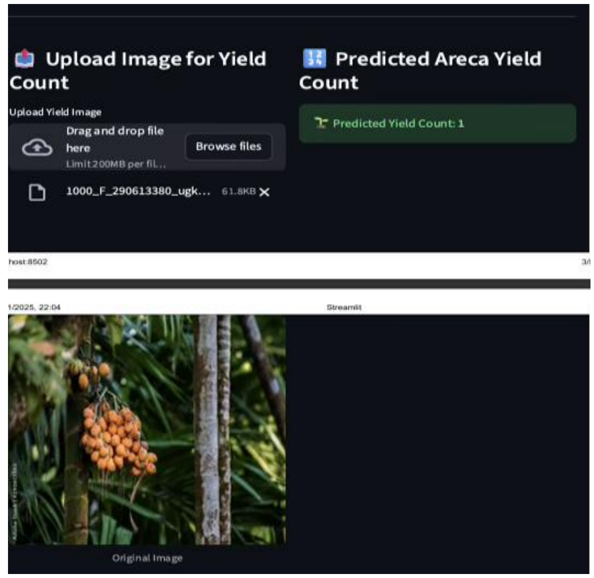

Fig. 5: Example of Yield Count Generated from Detection Output

- Yield estimates were accurate even when clusters overlapped, thanks to the combined use of segmentation area and direct object counting.
- Results were consistent across images captured by different camera settings and at varying flight altitudes, suggesting the approach generalizes reasonably well.

D. Arecanut Ripeness Detection API

Experiment: Images were submitted to the /predict/ and /predict/totalyield/ endpoints to test the API's end-to-end response.

Result:

- The API responded quickly in all test cases, returning structured JSON output with no noticeable latency for typical image sizes.
- Each response included the image filename, the total detection count, and the final estimated yield, giving users everything, they need in a single call.
- Testers found the interface straightforward to work with, and the real-time response made it practical for use during an active drone flight.

E. Disease Detection

Experiment: The disease detection model was evaluated on drone images containing both healthy and infected plant material.

Result:

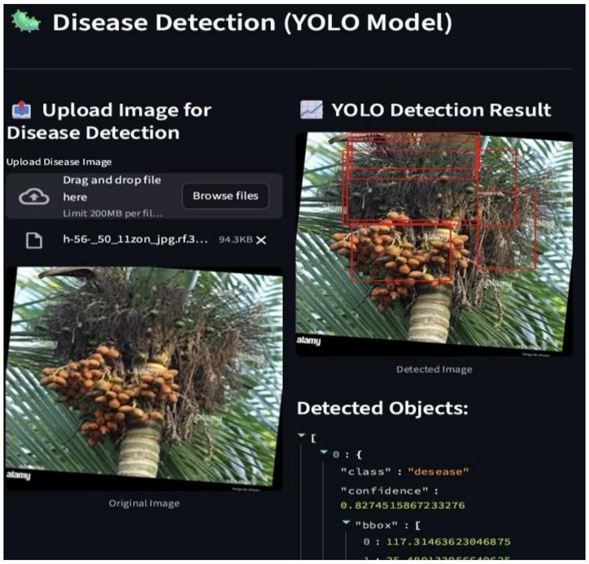

Fig. 6: Output Showing Detected Disease with Bounding Boxes

- Diseased regions were flagged accurately, with bounding boxes tightly fitted around the affected areas.
- Healthy portions of the plantation were consistently left unmarked, keeping the false-positive rate low.
- The model ran fast enough to support real-time monitoring during flight, which is exactly what field use requires.

VI. CONCLUSION

What started as a question — can a drone and a few deep learning models do what a trained agronomist does on foot? — turned into a system that, by most measures, answers yes. The framework we built successfully monitors arecanut plantations from the air, detecting inflorescences, classifying disease, estimating yield, and delivering results through a clean API and dashboard. The key outcomes are summarized below.

- Accurate Arecanut Detection:

YOLO consistently located clusters across varied lighting, density, and occlusion conditions.

- Effective Segmentation:

U-Net cleanly delineated arecanut regions, providing reliable inputs for both navigation and counting.

- Reliable Yield Estimation:

The hybrid counting approach produced consistent yield figures even when clusters overlapped.

- Real-Time Disease Monitoring:

The disease model identified infected areas quickly enough for use during live drone flights.

- User-Friendly API Interface:

Predictions were returned in structured JSON with low latency, making integration straightforward.

- Robust System Integration:

The drone, detection pipeline, and backend operated together stably throughout all field tests.

VII. FUTURE WORK

The current system works well for the conditions it was designed around, but there is clearly room to grow. A few directions stand out as particularly worthwhile:

- Broaden the disease model to cover additional conditions beyond Mahali Koleroga and stem bleeding, and add more granular ripeness categories.
- Explore multi-spectral and hyperspectral cameras, which can reveal stress signatures invisible to a standard RGB sensor.
- Develop smarter flight path algorithms that prioritize areas of likely interest, reducing battery consumption and flight time.
- Add a forecasting layer that uses historical detection data to predict yield trends and flag disease risk before outbreaks become visible.
- Move inference on-device by deploying lightweight model variants directly on the drone, cutting dependence on a wireless connection and reducing end-to-end latency.
- Close the loop by coupling the detection output with a precision spraying mechanism, so the same flight that identifies disease can immediately treat it.

REFERENCES

[1] R. Jayaram, P. Kumar, and S. Rao, “Design and Implementation of Agricultural Drone for Areca Nut Farms,” *International Journal of Agricultural Robotics*, vol. 12, no. 2, pp. 45–58, 2024.

[2] Y. Wang, L. Chen, and H. Zhao, "Development of an Autonomous Drone Spraying Control System Based on the Coefficient of Variation of Spray Distribution," Precision Agriculture, vol. 25, pp. 102–119, 2024.

[3] S. S. Prasanna, K. R. Anusha, R. Shreya, and K. Manjunath, "A Customized CNN for Arecanut Disease Detection," Journal of Smart Agriculture, vol. 5, no. 1, pp. 33–45, 2025.

[4] P. Keerthana, S. Spoorthi, R. Meghana, and A. R. Prakash, "Arecanut Plant Disease Detection System Using Deep Learning," International Journal of Intelligent Agriculture, vol. 8, pp. 89–101, 2025.

[5] Q. Qin, Y. Liu, and H. Zhang, "Ag-YOLO: A Real-Time Low-Cost Detector for Precise Spraying," Computers and Electronics in Agriculture, vol. 190, Art. no. 106392, 2021.

[6] K. Raj, A. M., S. Saravanan, and V. Muthukumar, "Artificial Intelligence-Based Autonomous Pesticide Spraying Drone Management for Sustainable Agriculture," Journal of Agricultural AI Systems, vol. 6, no. 2, pp. 55–71, 2025.

[7] P. K. Pratham, R. S., and M. V., "Autonomous Pesticide Spraying and Disease Detection UAV Platform for Areca Nut and Pepper Plantations," International Journal of Precision Agriculture, vol. 10, no. 3, pp. 120–136, 2022.