napari

File View Layers Plugins Window Help

layer controls

opacity: .52
blending: translucent
label: 1
brush size: 10
color mode: auto
contour: 0
n edit dim: 2
contiguous:
preserve labels:
show selected:

layer list

GH013060-cropped - Pecan
GH013060-cropped
GH013059-crop...nside ellipse
GH013059-crop...can - ellipse
GH013059-cropped - Adjusted
GH013059-cropped - Pecan
GH013059-cropped
GH013058-crop...nside ellipse
GH013058-crop...can - ellipse
GH013058-cropped - Adjusted
GH013058-cropped - Pecan
GH013058-cropped






-3 29 | 189

use <7> for transform, use <1> for activate the label eraser, use <2> for activate the paint brush, use <3> for activate the polygon tool, use <4> for activate the fill bucket, use <5> for pick mode

activity

napari

File View Layers Plugins Window Help

layer controls

opacity: .70
blending: translucent
label: 1
brush size: 10
color mode: auto
contour: 0
n edit dim: 2
contiguous:
preserve labels:
show selected:

layer list

20250808_14014...justed - Pecan
20250808_1401...1Z - adjusted
20250808_1401...-061Z - Pecan
20250808_1401...18-01-48-061Z

Layer list

-3 119 | 199

20250808_140146_manual_camera2_2025-08-08T18-01-48-061Z - Pecan [111 484 -218]:

activity

napari

File View Layers Plugins Window Help

layer controls

opacity: .50
blending: translucent
label: 1
brush size: 10
color mode: auto
contour: 0
n edit dim: 2
contiguous:
preserve labels:
show selected:

layer list

20250808_14014... [1] - tracked
20250808_1401... YOLO Seg [1]
20250808_1401...18-01-48-061Z

Layer list

Tracking complete: 1853 track(s) across 192/200

-3 62 | 199

use <7> for transform, use <1> for activate the label eraser, use <2> for activate the paint brush, use <3> for activate the polygon tool, use <4> for activate the fill bucket, use <5> for pick mode

activity

Performance & Metrics

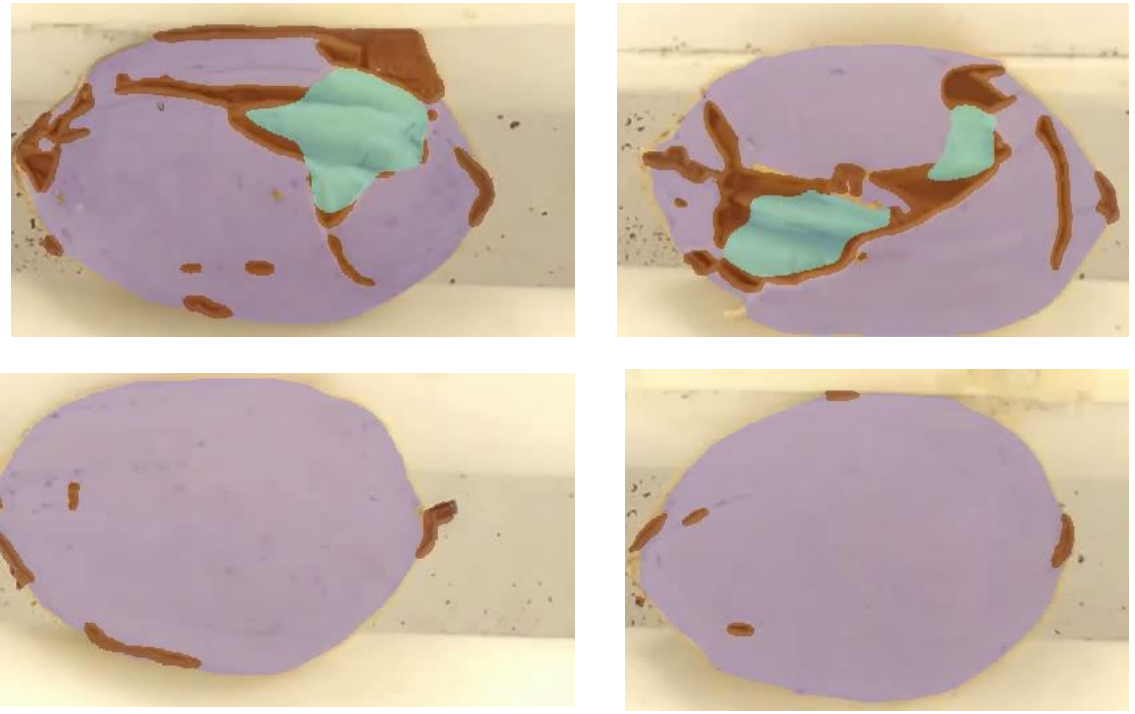
Item	Value
Model	YOLOv8n-seg (from yolov8n-seg.pt)
Class	Pecan only
Data	4 videos, 925 frames
Split	740 train / 185 val (80/20 by frame)
Train	20 epochs, batch 4, image size 640
Optimizer	AdamW (auto), AMP, GPU (RTX 3070 Laptop)
Time / size	~12 min, 6.8 MB, 3.3M params
Speed	**4.1 ms** inference / image (~106 FPS with pre/post)

Metric	Precision	Recall	mAP50	mAP50-95
Mask	0.913	0.902	0.937	0.743
Box	0.917	0.894	0.931	0.797

Performance & Metrics

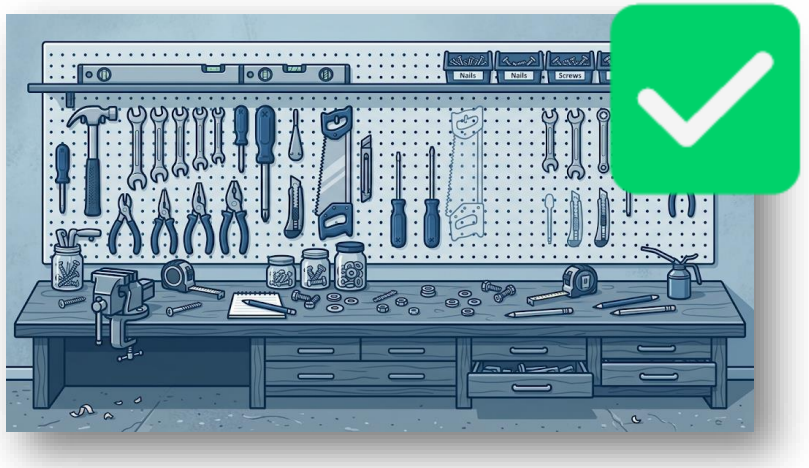
Item	Value
Model	Cascaded U-Net++ (MobileNet-V2 encoder)
Classes	Pecan, Crack, Kernel
Hierarchy	Pecan → Crack, Kernel (siblings under pecan)
Data	11 videos, **2661** frames
Split	**2129** train / **532** val (80/20 by frame)
Val load	532 frames; Kernel present in **240** / 532
Train	20 epochs, batch 4, image size **384**
Optimizer	AdamW, lr 0.001, GPU (RTX 3070 Laptop)
Time / size	**~1 h 45 min**, **82.9 MB**, **20.6M** params

Metric	Pecan	Crack	Kernel
IoU	0.962	0.821	0.682



Out-of-the-box agricultural reach

PecanPy is a multi-crop phenotyping & post-harvest QA platform, not a single-nut tool



Crop area	What you can look for
Nuts & hard-shell	Shell damage, splits, kernel exposure, foreign material
Tree & soft fruit	Bruises, rot, stem-end defects, size/shape
Vegetables & tubers	Surface defects, green shoulders, packing-line damage
Grains & pulses	Breaks, discoloration, mold
Pathology & breeding	Leaf spots, mildew, scars

Summary

Hardware Integration
- Retrofitting the processing line with dedicated camera tracking

Dataset Collection
- Recording comprehensive experimental runs under varying conditions

Training CV Models
- Developed the PecanPy workstation. Automated label generation
- Trained robust models

Pave the way for

Analysis
- Extract deep metrics & find parameter interactions

Closed-Loop Automation
- Leveraging vision and statistical models to monitor output and adjust machine parameters in real-time

Key Takeaways

For Research:

- Automatically generate annotations for datasets, reducing time and cost associated with manual annotation.
- PecanPy is suitable for users with little to no expertise in ML/DL.

For Agriculture:

- Computer vision integration is feasible for in-line inspection.
- Is an accurate, fast, reliable, and objective method, versus traditional human visual inspection, which is slow, and hard to standardize.

Publications and Knowledge Dissemination

Publication Possibilities:

- *[Preparing]* ***PecanPy*** in the ***Journal of Open-Source Software (JOSS)***
- *Vision-based multi-class detection of pecan shell cracks and kernel exposure for crack-dynamics analysis*

All slides available:
https://pecanresearch.uga.edu/

Thank you!

Questions & Comments