UF | IFAS Extension
UNIVERSITY of FLORIDA

UF/IFAS PEER REVIEWED AE616
doi.org/10.32473/EDIS-AE616-2026

# PhenoSnap: An AI-Powered Web Application for Automated Specialty Crop Trait Extraction[1]

Santhi Daggubati, Xu Wang, Xue Zhou, Shubham Singh, and Jessica Chitwood-Brown[2]

## Summary

Manual quantification of specialty crop traits, such as flowers and fruits, is often labor-intensive, time-consuming, and inconsistent, limiting scalability and precision. We present PhenoSnap, an artificial intelligence (AI)-powered web application that provides an intuitive and efficient interface for automated specialty crop trait extraction from images. Currently, PhenoSnap employs three deep learning models to quantify strawberry runners, mature and immature fruits, and flowers, as well as tomato fruits and flowers. Users can select different prediction models for image analysis and visualization, then download predicted results from the web interface for further analysis, such as yield estimation. PhenoSnap bridges the gap between advanced computer vision technologies and practical agricultural applications by eliminating the need for programming expertise. This ready-to-use solution can enable growers, breeders, and Extension faculty to accelerate field work and enhance decision-making related to strawberry and tomato yield estimation for breeding selections and strawberry runner management.

## Background

High-throughput plant phenotyping — the rapid, automated measurement of plant traits across large populations — is essential for accelerating crop improvement, yet traditional approaches often require expensive imaging systems, complex computational workflows, and advanced technical expertise (Wang et al. 2026). As a result, many breeding and research programs face barriers to adopting AI-driven phenotyping tools. To address these challenges, the Plant Phenomics Lab at the University of Florida (UF) Institute of Food and Agricultural Sciences (IFAS) Gulf Coast Research and Education Center (GCREC) developed PhenoSnap, a free, web-based application for plant image analysis. PhenoSnap runs entirely through a web browser (e.g., Firefox, Google Chrome, and Microsoft Edge) and is accessible on any internet-connected device, including cell phones, tablets, and laptop or desktop computers, making deep learning-based image analysis for plant phenotyping broadly accessible to breeding programs and researchers regardless of their technical resources. It removes key adoption barriers, because no programming expertise, local software installation, or specialized computing hardware is required.

The system builds on recent advancements in computer vision, cloud-based computing, and open-source artificial intelligence (AI) frameworks. By leveraging pretrained models trained on thousands of manually labeled ground-based and drone-based images, PhenoSnap can automatically detect and quantify morphological components such as flowers, immature fruits, mature fruits, and runners in strawberry, as well as flowers and green fruit in tomato. Users can simply upload images of strawberry or tomato plants through the web interface. The web app then performs image analysis and visualization using the UF Research Computing infrastructure to provide support for fruit yield estimation and crop management.

Unlike conventional software that requires local installation and manual configuration, PhenoSnap operates entirely through a browser, making it accessible to users across diverse agricultural programs. The platform contributes to the broader UF/IFAS mission of translating cutting-edge research into practical, accessible tools that enhance crop management, breeding efficiency, and scientific collaboration.

## Functional Overview of PhenoSnap

PhenoSnap was developed with the central goal of making advanced deep-learning-based phenotyping accessible to researchers and practitioners without programming or computational expertise. Deep learning is a type of artificial intelligence that learns patterns from large sets of example data, such as images, to automatically recognize and measure features of interest. Its design emphasizes simplicity and usability, integrating every stage of the image analysis pipeline (uploading, model selection, inference, visualization, and data export) within a unified web interface.

PhenoSnap is hosted on UF's Research Computing (UFRC) infrastructure and can be accessed securely through a web browser on any modern device (e.g., desktop computers or cell phones) at https://phenosnap.rc.ufl.edu. Upon visiting the site, users land on a simple home interface with a left sidebar for navigation and a main workspace that outlines the step-by-step workflow (Model Selection > Upload Images > Run Inference > Visualization > Results Export) with Run Inference, Rerun, and Clear controls (Figure 1). All computation relies on backend computing resources on the UF HiPerGator high-performance computer. No local installation or configuration is required. This browser-based deployment ensures compatibility across operating systems and enables multiple concurrent users to run independent phenotyping tasks in parallel. For mobile users, PhenoSnap's responsive design supports smartphones and tablets without any additional setup. Growers and field scouts, for example, can photograph crops directly in the field, upload images on the spot (with internet), and receive results within the same session, enabling rapid, on-site trait evaluation without returning to a desktop workstation.

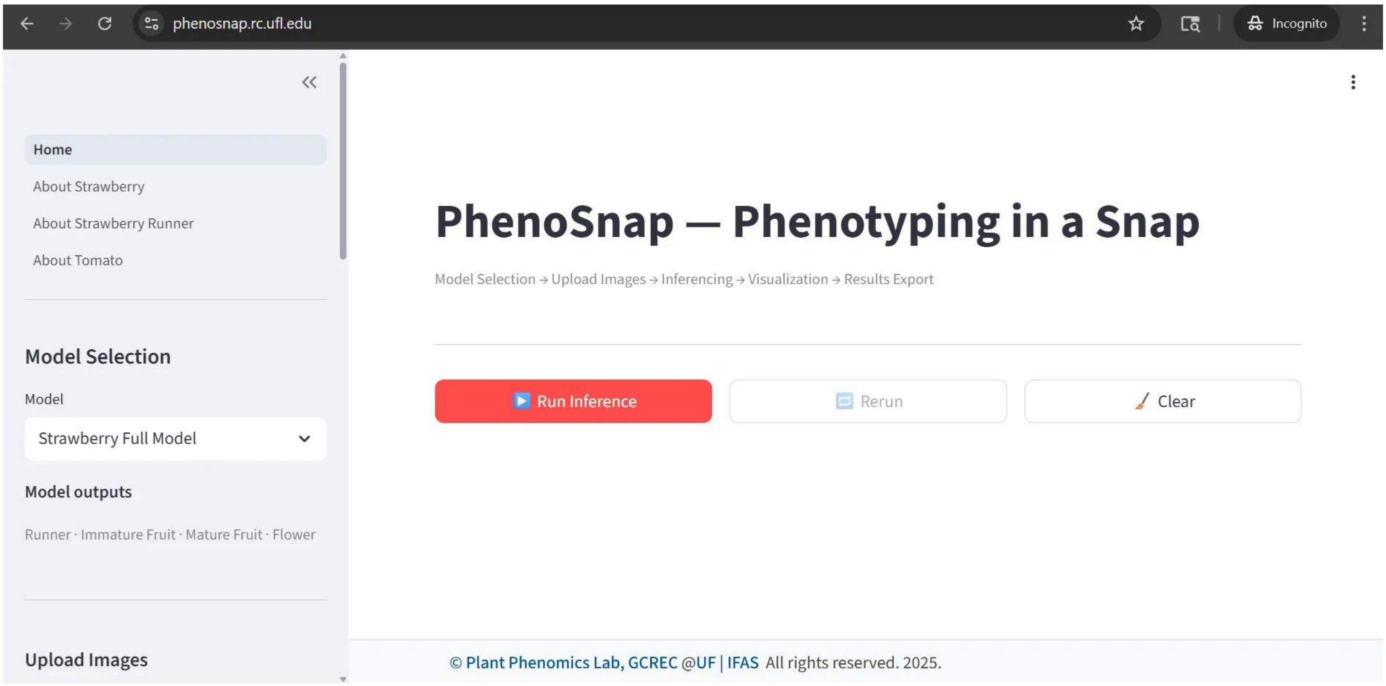



Figure 1. PhenoSnap home page (desktop view).
Credit: Santhi Daggubati, UF/IFAS.

The workflow (Figure 2) begins with **Model Selection** on the left navigation bar, which lists all currently supported deep learning models and their associated crop types. Each model is accompanied by a description detailing its detecting objects (e.g., flowers, fruits, runners). Following model selection, the user proceeds to the **Upload Images** step, which accepts RGB images of related crops. Users can upload one or multiple images in JPG (or JPEG) and PNG formats. Upon uploading, PhenoSnap automatically prepares the images for analysis by adjusting their size and formatting so they can be processed by the selected model. The original image files are not changed or permanently stored. Instead, temporary working copies are created only for the duration of the user's session to generate predictions based on the settings selected in the **Advanced Settings** panel.

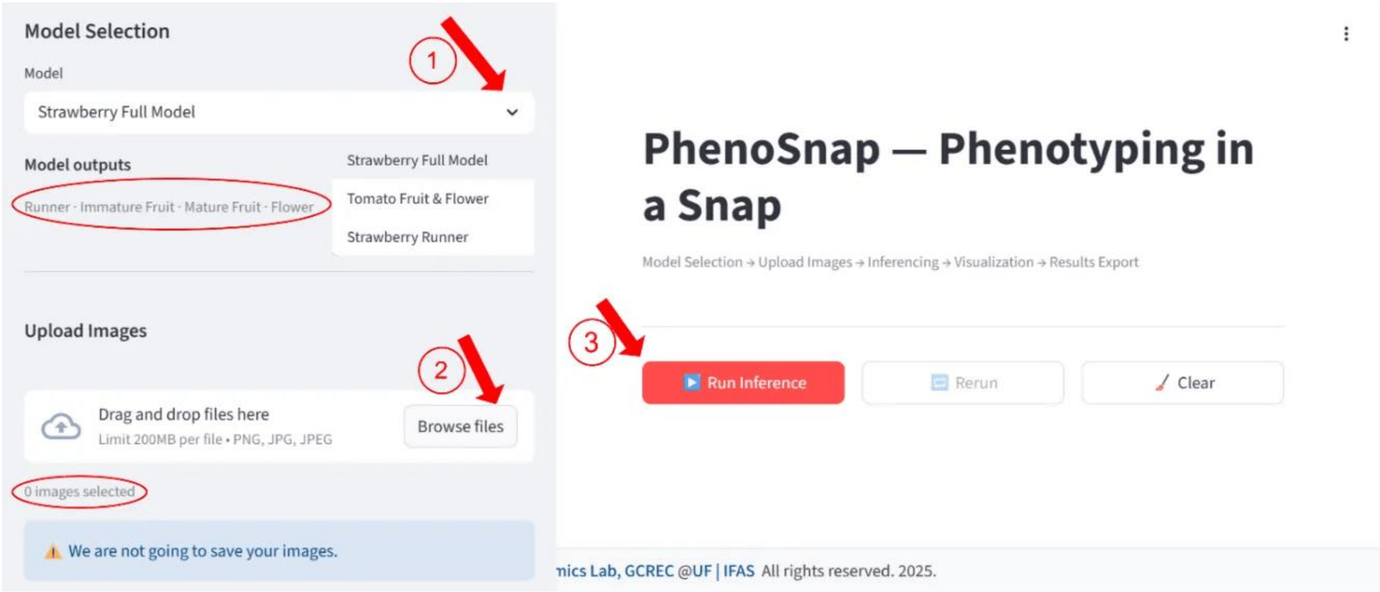


Figure 2. General workflow of PhenoSnap: (1) Model Selection, (2) Upload Images, and (3) Run Inference. Credit: Santhi Daggubati, UF/IFAS.

When users click **Run Inference**, selected images will be processed by the deep learning model chosen. Inference is the step where the trained model uses what it learned during training to examine a new image and make predictions, such as identifying and counting plant parts. A progress indicator (the blue bar at the bottom of Figure 3) appears while processing runs. PhenoSnap currently runs on the UFRC infrastructure, and a single image processing typically completes in 3–5 seconds. When inference is completed, PhenoSnap populates a results table summarizing each class and its total count for all the uploaded images and renders predictions as overlays on the original images (Figure 4). The rendered output images have a per-instance boundary box and mask with a predicted class name and confidence value. Users can export outputs by clicking **Download all outputs (ZIP)** at the bottom of the page. The archived output includes per-image prediction files, per-image class counts, and a consolidated master CSV combining class counts across all images (Figure 5).

Figure 3. Inference pipeline status. After "Run Inference," images are preprocessed and sent to the selected model. A blue progress bar indicates progress until results are ready.
Credit: Santhi Daggubati, UF/IFAS.

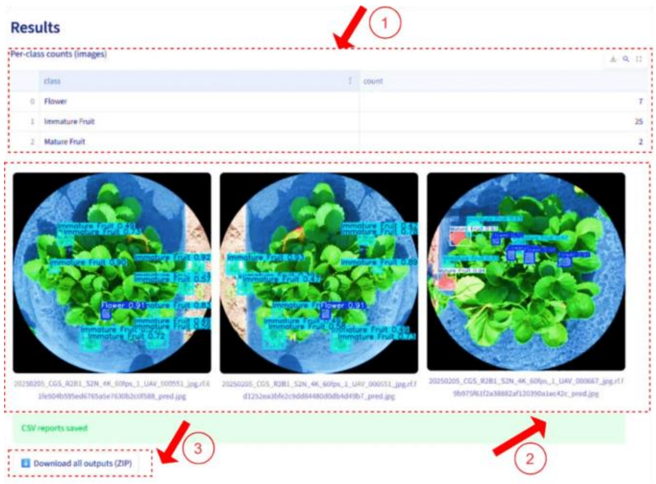


Figure 4. Results visualization and exporting. (1) Results table summarizing per-class (e.g., flower, immature fruit, and mature fruit) counts. (2) Annotated outputs (masks/boxes) of each plant part in uploaded images. (3) Download button to export annotated images and CSV files.
Credit: Santhi Daggubati, UF/IFAS.

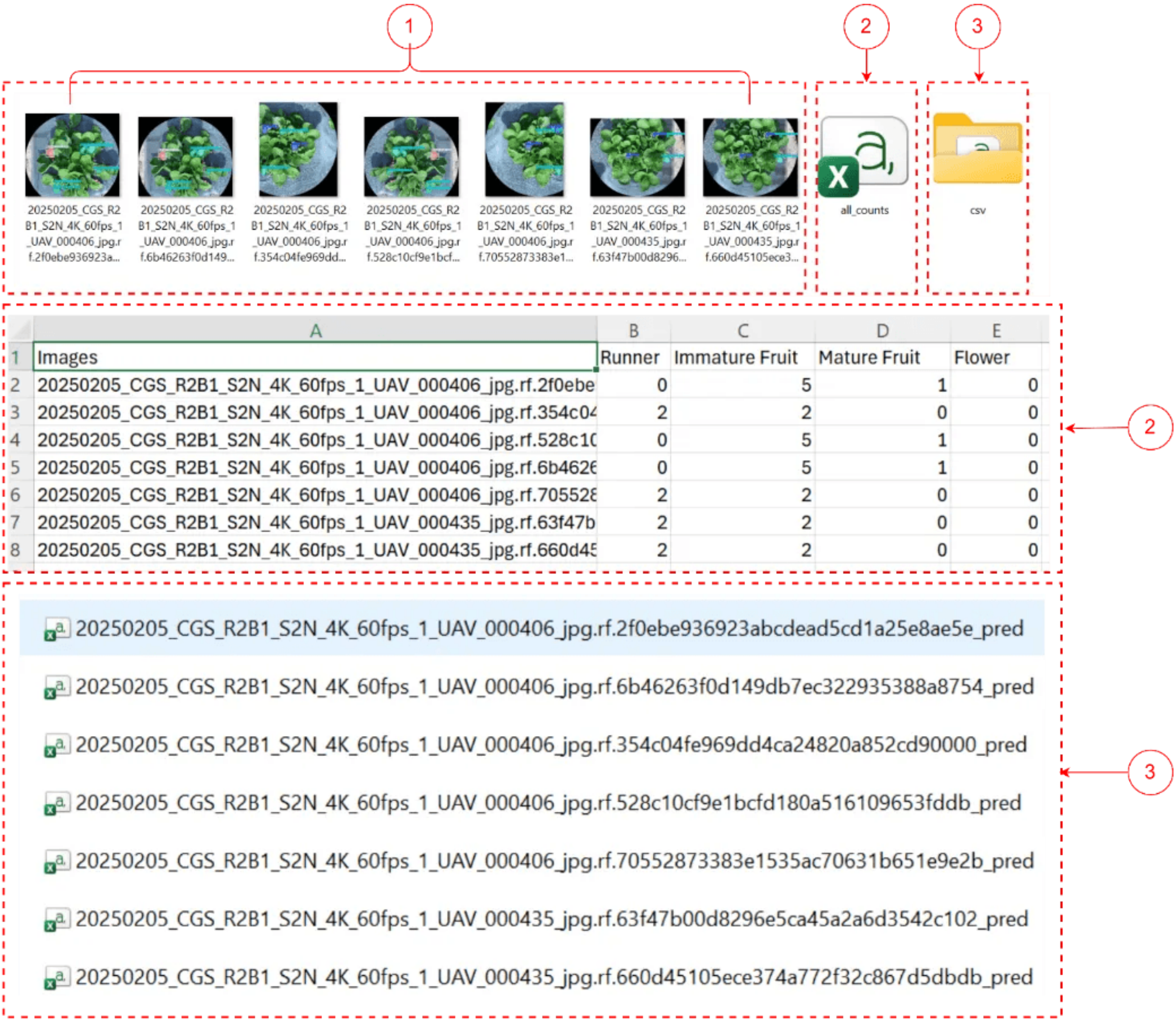



Figure 5. Example of exported PhenoSnap outputs, including (1) annotated images, (2) a consolidated summary CSV file, and (3) per-image class counts.
Credit: Santhi Daggubati, UF/IFAS.

## Deep Learning Models

PhenoSnap is powered by crop-specific deep learning models to identify and quantify key morphological organs in specialty crop imagery. In the current release, PhenoSnap supports three production models (Table 1): (1) Strawberry Full Model, (2) Tomato Fruit and Flower, and (3) Strawberry Runner. For additional details on training data, exemplary images, training procedures, and full results, users can check the **About** pages accessible from the top navigation bar in PhenoSnap (Figure 6).

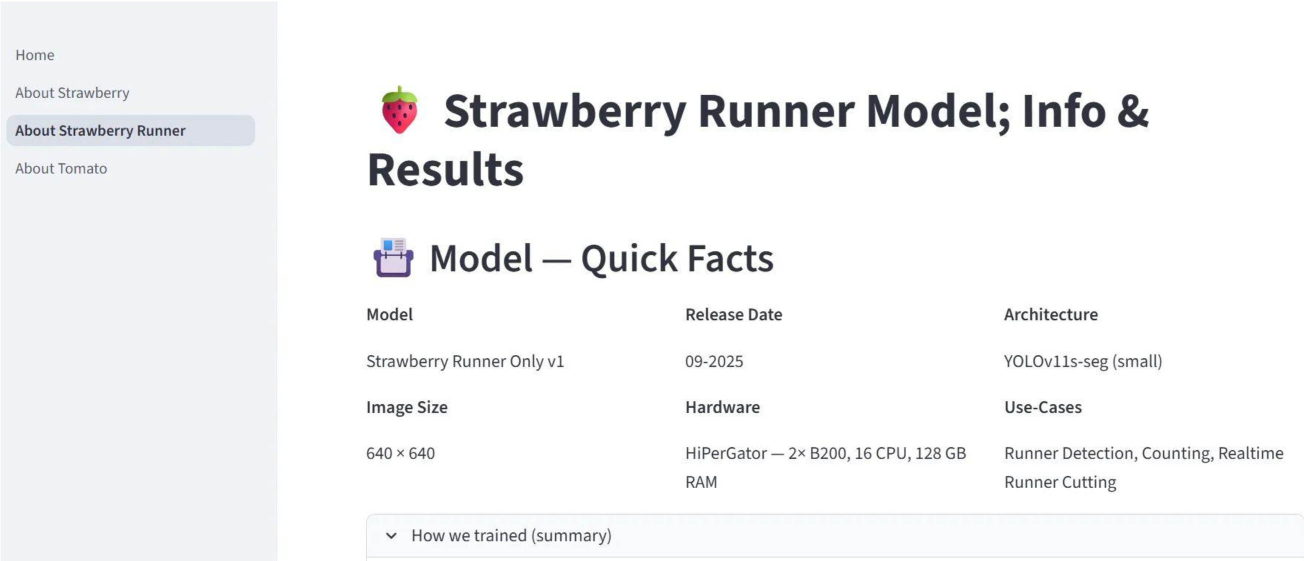



Figure 6. Model information page: “About Strawberry Runner” in PhenoSnap.
Credit: Santhi Daggubati, UF/IFAS.

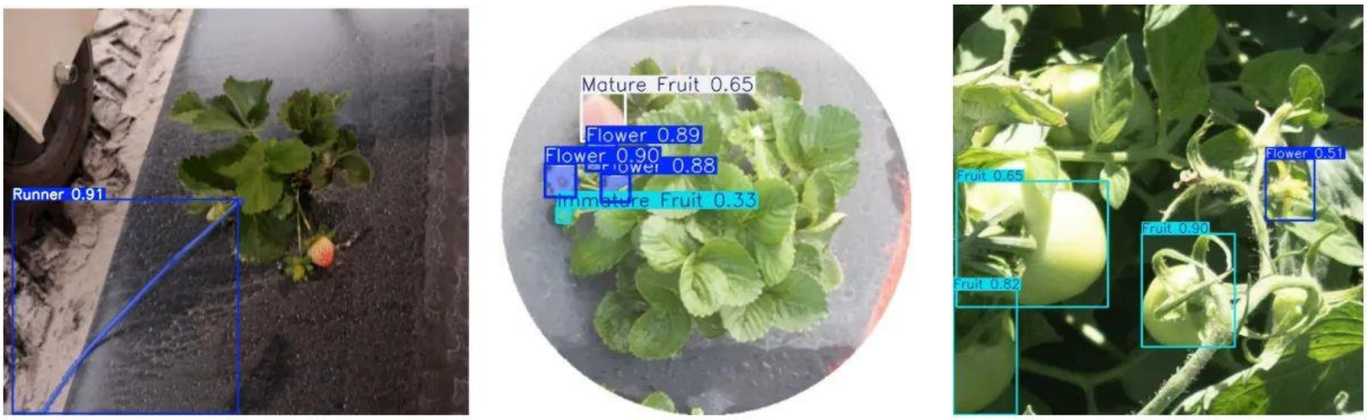
Table 1 shows the following.

- **Deep Learning Architecture**: The type of artificial intelligence (AI) model used to analyze images and recognize plant parts. Different architectures vary in size, speed, and accuracy. PhenoSnap uses the YOLO (You Only Look Once) model family, with different model sizes applied across the three supported models.
- **Object Detected**: The specific plant features the model is trained to identify and count, such as runners, flowers, or fruits.
- **Training Images**: The number of labeled images used to teach the model what different plant parts look like.
- **Validation Images**: A separate set of images used to test the model’s performance during development and ensure it works on new, unseen images.
- **F1-score**: A performance metric that summarizes how accurately the model identifies objects, balancing missed detections and false detections. Values closer to 1 indicate better performance.
- **mAP-50 (mean Average Precision at 50%)**: A standard accuracy metric in computer vision that measures how well the model detects objects with sufficient overlap between predicted and actual locations. Higher values indicate better detection accuracy.
- **Model Release Time**: The month and year when the model was finalized and integrated in PhenoSnap.

The **Strawberry Runner** model was trained to automatically detect and segment runners in strawberry plants (Figure 7a). The dataset includes over 10,000 manually annotated field images collected during multiple growing seasons from 2022 to 2025. Dataset preparation and the training process are detailed in Zhou et al. (2025a), and a subset of the dataset has been publicly released on Dryad (Zhou et al. 2025b).

The **Strawberry Full Model** extends the runner detection model to recognize multiple parts in strawberry plants, including flowers, immature fruits, mature fruits, and runners (Figure 7b). It was trained on an expanded dataset that combines the subset of the original runner dataset with additional manually annotated images collected across diverse flowering and fruiting stages over seasons from 2022 to 2025.

The **Tomato Fruit and Flower** model was developed to detect flowers and green fruits in tomato plants with compact growth habits (Figure 7c). The dataset consists of over 2,000 annotated images captured from field trials at UF/IFAS GCREC under varied lighting and canopy structures. Unlike the other two segmentation-based models, this model was trained exclusively for object detection of flowers and fruits (Singh et al. 2025).




(a) Strawberry Runner model (b) Strawberry Full model (c) Tomato Fruit & Flower Model

Figure 7. Example outputs produced by PhenoSnap for each supported model. (a) Strawberry Runner model. (b) Strawberry Full Model. (c) Tomato Fruit and Flower model.
Credit: Xu Wang and Shubham Singh, UF/IFAS.

## Advanced Settings

PhenoSnap provides adjustable parameters that allow users to optimize the deep learning models' prediction results according to specific image characteristics and project needs. Adjusting these parameters can help improve prediction quality, although the impact may vary depending on the dataset.

- **Device:** The runtime device can be set to ***auto***, ***gpu***, or ***cpu***. In ***auto*** mode (default), the application preferentially schedules inference on an available graphics processing unit (GPU) and falls back to a central processing unit (CPU) otherwise. A GPU is designed to perform many calculations in parallel and is generally faster for image analysis, while a CPU performs calculations sequentially and is typically slower for this type of task; ***auto*** mode automatically selects the fastest available option. Explicit selection is helpful for troubleshooting or for benchmarking GPU vs. CPU latency. One NVIDIA L4 GPU is reserved on UFRC for GPU-based inferencing.
- **Confidence threshold:** This setting controls the model's confidence before it decides that something in the image is a final object. The confidence value goes from 0 to 1, where 1 means the model is entirely sure. A lower threshold lets the model count more objects but may also include incorrect detections (Figure 8a). A higher threshold makes the model pick only objects it is sure about, which means fewer detections but usually more accurate ones (Figure 8b). Minor changes (for example, adjusting by 0.05–0.10) are recommended. Unless otherwise stated, the default confidence threshold value is 0.25.

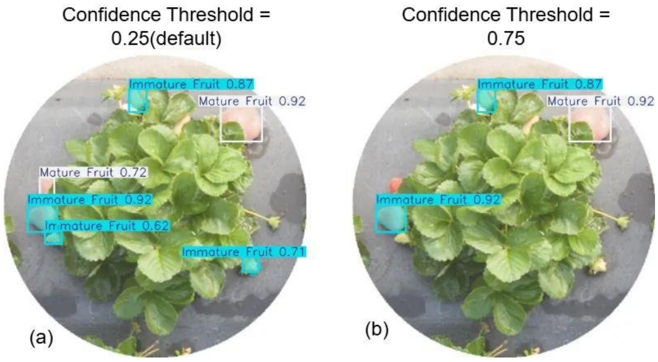



Figure 8. Comparison of prediction results using different confidence thresholds. (a) Confidence threshold of 0.25, with six total detections. (b) Confidence threshold of 0.75, with three total detections.
Credit: Santhi Daggubati, UF/IFAS.

- **IoU threshold:** The Intersection-over-Union (IoU) setting controls suppression of overlapping detections in the output. Overlapping or duplicate detections occur when two prediction boxes or masks partially cover the same plant part, such as closely spaced fruits or flowers, causing the model to decide whether they represent one object or multiple objects (Figure 9). Higher values are more conservative and may remove adjacent, valid instances in crowded scenes, while lower values retain more neighbors but can yield duplicates. The default value of 0.5 matches the value used during model validation. To prevent the model from merging adjacent duplicate detections, users can lower this value; otherwise, increase it.

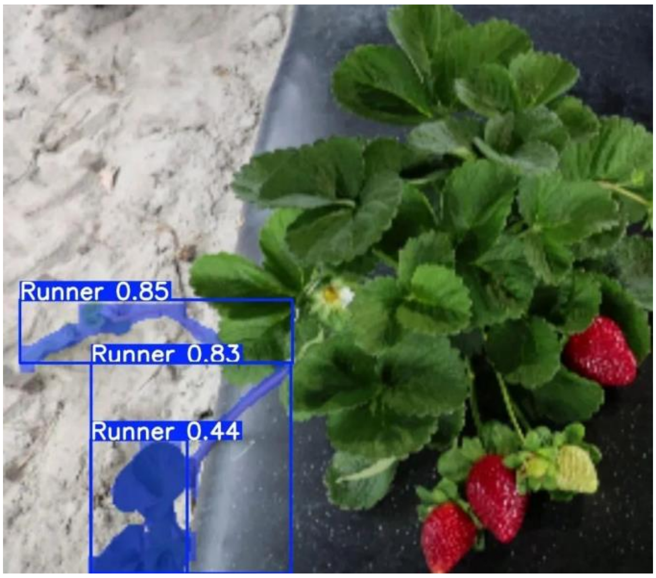



Figure 9. Overlapping or duplicate detections of the strawberry runner.
Credit: Xue Zhou, UF/IFAS.

- **Image size:** Before analysis, images are resized to a default size (640 x 640 pixels). Larger size values reveal more details and smaller targets expected to be detected (e.g., fruits or flowers in high-altitude aerial images). However, the prediction will take longer to process and use more memory. Smaller size values will result in faster prediction but may miss tiny objects.
- **Batch size (images):** The size value controls the number of images processed concurrently. For most users, the default setting is sufficient and does not need adjustment, unless large numbers (e.g., 100) of images are processed at once. Increasing batch size can speed up predictions on large image sets, but may increase the risk of “out of memory” errors. Batch size does not change model weights but only affects runtime efficiency. The default batch size is set to 8, while the initial models are trained in a batch size of 64.

We recommend users begin with default settings (Figure 10). If the objects detected in images seem undercounted, users can slightly lower the **confidence threshold.** If the outputs appear noisy or duplicated in clusters, raising the **IoU threshold** may improve the prediction.

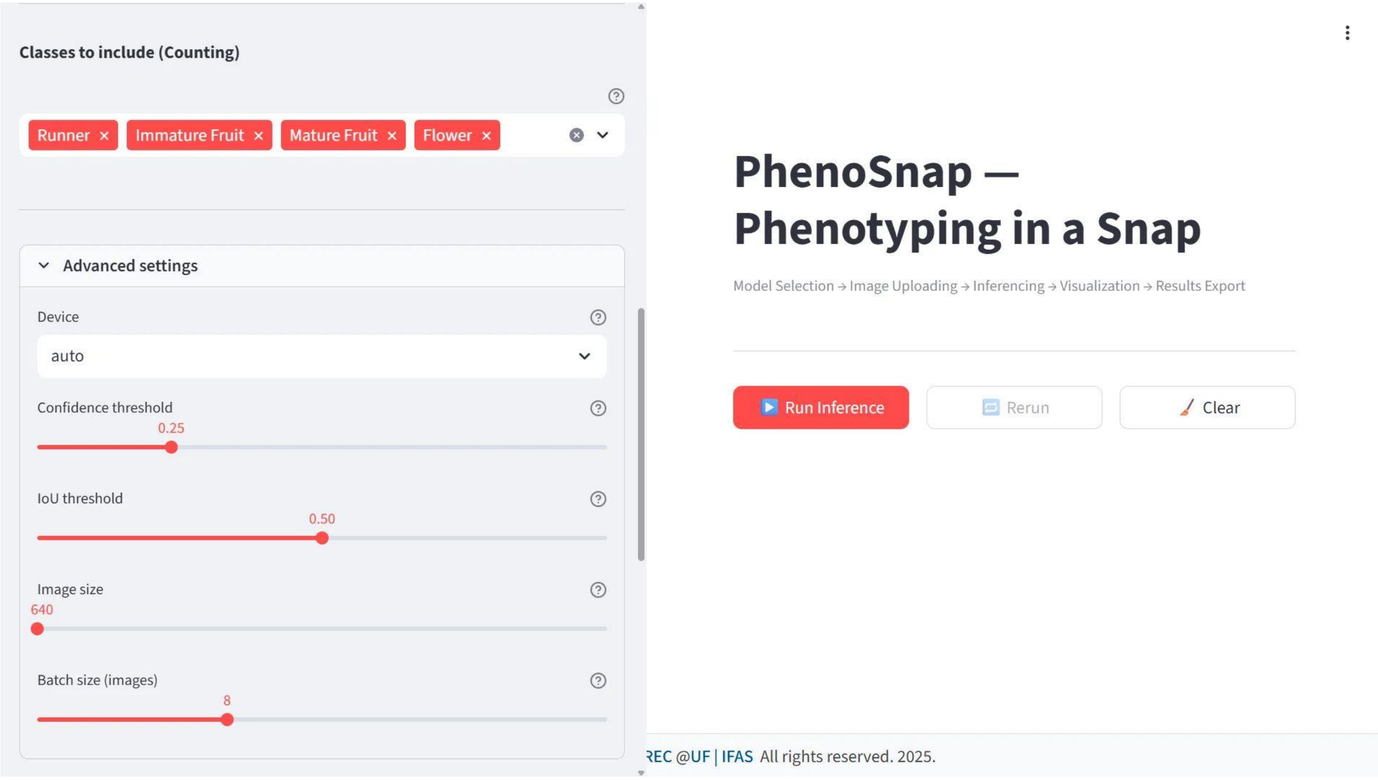



Figure 10. Default values for Advanced Settings for PhenoSnap inferencing.
Credit: Santhi Daggubati, UF/IFAS.

## Other Features

PhenoSnap also includes other features that enhance usability and support users’ feedback (Figure 11).

- **Classes to include (Counting):** A class in PhenoSnap refers to a specific type of plant part that the model is trained to recognize, such as a strawberry flower, fruit, or runner. Users can select counting to a subset of classes (e.g., runners only, or flowers + fruits). This setting filters only the per-class summary table, ensuring that exported CSVs only include the selected classes to predict. Exclusion does not alter the underlying model. It is a post-inference selection for the export results.

- **Total inferences:** The interface displays a running count of clicks processed through the **Run Inference** button across all the sessions and all the users. It provides information on how many predictions the model made from deployment to the current time.
- **Rate PhenoSnap:** A lightweight rating widget allows users to provide qualitative feedback after completing inferences. While this does not influence model predictions, it helps our lab with the performance of the models and further improvements.

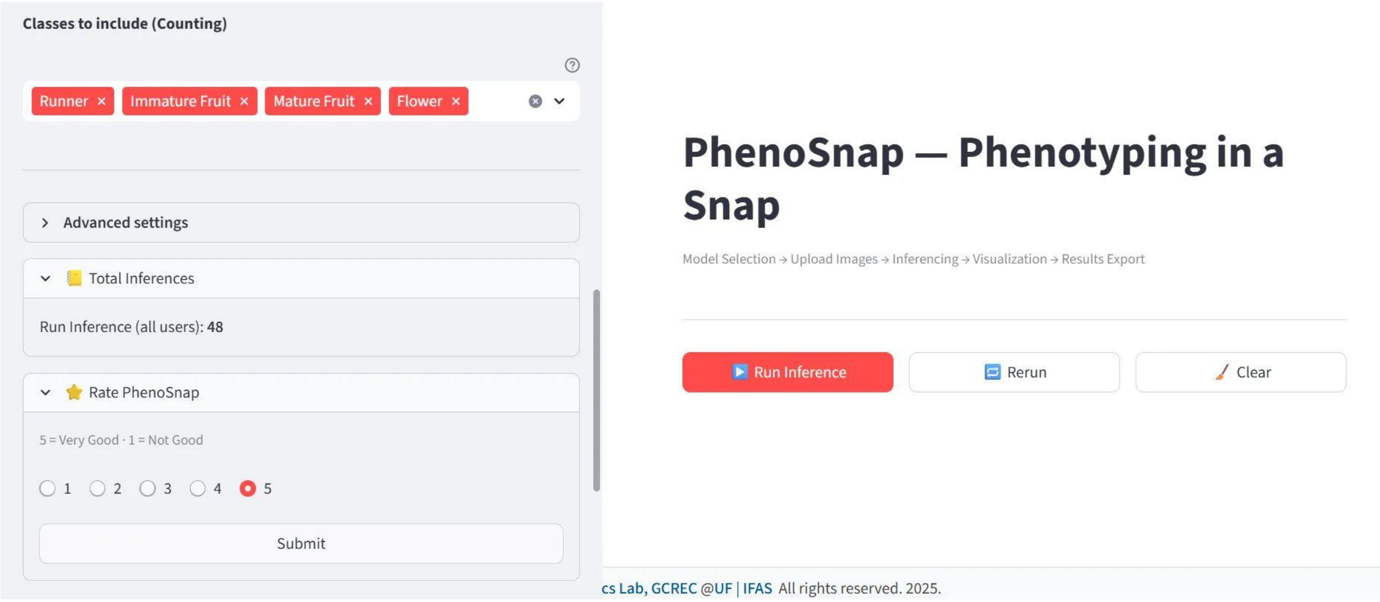


Figure 11. PhenoSnap interface showing class selection, total inference count, and user rating features. Credit: Santhi Daggubati, UF/IFAS.

## Limitations and Future Work

The current PhenoSnap release performs best on well-lit, in-focus RGB images (e.g., digital images taken by cell phones or cameras) in which target plant parts occupy sufficient pixel area. At the same time, severe shadows, glare, motion blur, or small objects make the application more likely to miss the prediction. Images used to train the deep learning models were mainly collected at UF/IFAS GCREC. Therefore, the models have only learned what fields typically look like (bed layout/color, ground color, lighting, and camera angles) in a limited background environment. With photographs showing different conditions, new cultivars, other environmental styles, different backgrounds (bed layout), or unusual lighting, the deep learning models may not perform with the expected accuracy. For example, strawberry runner tips with large leaves packed tightly together can be mistaken for other plant parts, and very long and dense runner clusters can be hard to separate into individual detections. Additionally, the tomato model was trained as detection-only (i.e., only detects the presence and location of objects, but does not outline their boundaries), and exclusively on flowers and green fruits; consequently, red tomatoes will not be recognized by the model at this stage. Furthermore, small objects are difficult to detect in aerial images captured at high altitudes. Uploading plant-level or plot-level images cropped from large-scale, field-level aerial images before prediction may help improve prediction performance.

## Conclusion

PhenoSnap is a web-based application that allows breeders, growers, and Extension faculty members to upload crop images for quick counting of strawberry and tomato flowers and fruits and strawberry runners, supporting field assessments and yield estimation. Users can upload images, run AI-powered inference, visualize detection results, and download structured files for further analysis. PhenoSnap reduces labor and time costs in field measurements while improving the consistency of measurements. In the future, PhenoSnap will continue to grow by adding more crop types, improving model accuracy, and making the system even more reliable and user-friendly for crop research and field applications.

## Acknowledgments

This study was supported by two grants from the Florida Strawberry Research and Education Foundation for the years 2023 and 2024 (AGR00028990 and AGR00031844). The authors express their gratitude to Dr. Vance Whitaker, strawberry breeder and the leader of the UF Strawberry Breeding Program, and his research team for their assistance in supplying plant materials and maintaining the strawberry and tomato fields that made this study possible.

## References

Singh, S., D. Choi, J. Chitwood-Brown, and X. Wang. 2025, July 30. “FRBR 1 — Machine Vision for Detecting and Quantifying Fruits and Flowers to Evaluate Concentrated Fruit Set in Tomato." American Society for Horticultural Science (ASHS) Annual Conference, New Orleans, LA, United States. https://ashs2025.sched.com/event/247AF

Wang, X., D. Choi, D. Adams, et al. 2026. "Artificial Intelligence-Powered Plant Phenomics: Progress, Challenges, and Opportunities." *The Plant Phenome Journal* 9(1): e70060. https://doi.org/10.1002/ppj2.70060

Zhou, X., X. Wang, L. Ji, S. Daggubati, K. Shen, and V. M. Whitaker. 2025a. "Deep Learning for Strawberry Runner Detection Integrating Ground and Aerial Imaging." *Smart Agricultural Technology* 12: 101290. https://doi.org/10.1016/j.atech.2025.101290

Zhou, X., X. Wang, V. M. Whitaker, L. Ji, K. Shen, and S. Daggubati. 2025b. "Ground and Aerial Imagery Dataset for Strawberry Breeding Trials: Training Deep Learning Models for Runner Detection and Segmentation." Dryad. https://doi.org/10.5061/dryad.bzkh189nw

Table 1. Summary of crop-specific deep learning models implemented in PhenoSnap.

| Model | Deep Learning Architecture | Object Detected | Training Images | Validation Images | Object Segmentation/Detection Accuracy | | Model Release Time |
|---|---|---|---|---|---|---|---|
| | | | | | F1-score | mAP-50 | |
| Strawberry Runner | YOLOv11s-seg | Runner | 9,875 | 2,040 | 0.975 | 0.985 | September 2025 |
| Strawberry Full Model | YOLOv11x-seg | Flower, Immature Fruit, Mature Fruit and Runner | 8,796 | 2,932 | 0.926 | 0.949 | September 2025 |
| Tomato Fruit and Flower | YOLOv11x | Fruit and Flower | 2,079 | 525 | 0.850 | 0.895 | June 2025 |

[1] This document is AE616, a publication of the Department of Agricultural and Biological Engineering, UF/IFAS Extension. Original publication date June 2026. Visit the Ask IFAS website at https://ask.ifas.ufl.edu for the currently supported version of this publication. 

[2] Santhi Daggubati, M.S., applied data science, UF Department of Engineering Education, UF/IFAS Gulf Coast Research and Education Center; Xu Wang, assistant professor, phenomics for plant breeding, Department of Agricultural and Biological Engineering, UF/IFAS Gulf Coast Research and Education Center; Xue Zhou, postdoctoral associate, Department of Agricultural and Biological Engineering, UF/IFAS Gulf Coast Research and Education Center; Shubham Singh, graduate research assistant, Department of Horticultural Sciences, UF/IFAS Gulf Coast Research and Education Center; Jessica Chitwood-Brown, assistant professor, horticulture and tomato breeding, Department of Horticultural Sciences, UF/IFAS Gulf Coast Research and Education Center, Wimauma, FL; UF/IFAS Extension, Gainesville, FL 32611.