

Project Title: Granny - image based analysis of fruit quality; TR-24-100A

Report Type: Final Project Report Year 2

Primary PI: Loren Honaas
Organization: USDA ARS TFRL
Telephone: 509.433.1143
Email: loren.honaas@usda.gov
Address: 1104 North Western Ave
Address 2:
City/State/Zip: Wenatchee, WA 98801

Co-PI 2: Stephen Ficklin
Organization: WSU Department of Horticulture
Telephone: 509.335.4295
Email: stephen.ficklin@wsu.edu
Address: PO Box 646414
Address 2:
City/State/Zip: Pullman, WA 99164

Co-PI 3: Dr. Huiting Zhang
Organization: WSU Department of Horticulture
Telephone: 509.335.4295
Email: huiting.zhang@wsu.edu
Address: PO Box 646414
Address 2:
City/State/Zip: Pullman, WA 99164

Cooperators: Allan Brothers Inc., Stemilt Growers LLC, WA Tree Fruit Research Commission, Dr. Carolina Torres (WSU TFREC), Dr. Rachel Leisso (USDA ARS TFRL), Hectre, Pomona Hort Innovation

Project Duration: 2-Year

Total Project Request for Year 1 Funding: \$77,176

Total Project Request for Year 2 Funding: \$73,944

Other related/associated funding sources:

Awarded:

Funding Duration: annual congressional appropriation

Amount: \$85,000

Agency Name: USDA ARS

Notes: 3-year total \$255,000: Personnel \$180,000, Consumables/Supplies \$30,000, Equipment (including computational resources): \$45,000

Funding Duration: 2022-2024

Amount: \$46,000

Agency Name: Washington Tree Fruit Research Commission (Award #AP-22-101A)

Notes: The high-performance compute node that was requested as part of a supplemental for the AP-22-101 will be utilized for this project, thus aside from a desktop computer for the postdoc in this budget, no other computing equipment is necessary.

Funding Duration: 2017-2022

Agency Name: US National Science Foundation (NSF) Award #1659300

Amount: \$150,000

Notes: A portion of this award was used to fund 600 Terabytes of storage for execution of scientific workflows and storage of results. That infrastructure is still available and we will use that infrastructure for this project.

Budget 1

Primary PI: Loren Honaas

PI requests no funding

Budget 2

Co PI 2: Stephen Ficklin

Organization Name: WSU Department of Horticulture

Contract Administrator: Anastasia Mondy

Telephone: 509.335.6885

Contract administrator email address: anastasia.mondy@wsu.edu

Item	2024	2025
Salaries	\$50,738.00	\$52,768.00
Benefits	\$18,438.00	\$19,176.00
Wages		
Benefits		
RCA Room Rental		
Shipping		
Supplies ¹	\$6,000.00	
Travel	\$2,000.00	\$2,000.00
Plot Fees		
Miscellaneous		
Total	\$77,176.00	\$73,944.00

Footnotes: ¹machine learning capable desktop computer, camera equipment, imaging station supplies, other minor consumables

Budget 3

Co PI 3: Huiting Zhang

Co-PI Budget is together with WSU's Budget #2 above.

Specific project objectives:

- 1) Build a stable version of Granny that includes a user-friendly starch rating module.
- 2) Provide a robust starch rating module for the industry.
- 3) Build a data set to train and test the software.
- 4) Provide outreach and training.

Significant findings, results, and milestones from 2025:

To provide some context, In 2025 we brought on a new programmer as our previous programmer left for graduate school. This new programmer required training and an onboarding period. The following organizes significant findings, results, and milestones as accomplishments we performed in order of the four stated objectives above.

1. *Build a stable version of Granny that includes a user-friendly starch rating module*
 - a. **Previously stated objectives for 2025:**
 - i. Continue redesigning the software. Proper software design supports flexibility and reduces long-term maintenance costs. Taking the time now to do something correctly will save us headaches downstream. Our MVC framework design (as described below) is not yet complete and we will finish this in 2025.
 - ii. To meet software robustness standards we will continue to add a complete suite of functional tests. This will ensure that as we make changes, we don't accidentally break what works between releases. It ensures the robustness of the software.
 - iii. We will work to ensure the software can work on a Windows PC, currently, it only works on a Linux PC.
 - b. **What was accomplished:**
 - i. We met our goal of completing the software design. While there is always room for improvement and expansion, we consider the software design at a stable and complete state. The primary design objective has been met that rater modules should be plug-and-play. Developers (either us, or industry partners) only need to focus on writing a new analysis and not worry about how to integrate their new tool. Granny utilizes a design approach termed Model-Viewer-Controller (MVC) which fully separates into modules (that can easily be swapped in-and-out and replaced) the data model, the user interface (viewer), and the controller (analysis modules). New modules are integrated automatically so long as the development protocols are followed. The protocols for creating new analysis are described in the Granny online documentation https://granny.readthedocs.io/en/dev/dev_guide/adding_analysis.html. While our original objective was to create starch rating modules, The MVC framework allows Granny to be **scalar** and **modular**. It is scalar in that new analysis modules or user interfaces can be easily added without major changes to the overall software. Moreover, Granny was designed to meet FAIR (Findable, Accessible, Interoperable and Reproducible) data standards in that the design facilitates the Interoperability and Reproducible parts of FAIR. Granny supports interoperability in that all starch rating arguments and parameters are fully exposed, documented, and have strict and well documented metadata describing the parameters.
 - ii. We will meet our goal of meeting robustness standards. We have implemented a complete set of functional tests. These provide a battery of stress tests that we can use to check for problems with the software before we

make a release. In summary we have strong coverage of the underlying data types (parameters), good coverage of import and managing of Images and IO. We are missing some functional tests for our starch analysis module, but expect to have that completed by the time the project ends.

- iii. We redirected efforts on the Windows interface. As reported last year, in 2024 we began working with Meg Staton's group at the University of Tennessee to build a web-front end Graphical User Interface (GUI) for Granny rather than a windows interface. This will allow Granny to be used more broadly across UNIX, MacOS and Windows-based computers. This work by the Staton team was separately funded by a USDA project but with close integration of our efforts.
2. *Provide a robust starch rating module for the industry.*
 - a. ***Previously stated objectives for 2025:***
 - i. While tests look great so far, we will continue testing partner images (WTFRC, Stemilt, AllanBros) to ensure consistent and quality rankings.
 - ii. Granny currently converts its ratings into eight starch rating scales (e.g., Granny Smith, Jonagold, WA 38, etc.). We will be adding more starch rating scales.
 - b. ***What was accomplished:***
 - i. We continued to test images. Our team has continued to collect in-house images for the 2025 season and evaluate performance of Granny on those images. We have also tested images from the WTFRC WA 64 variety. Performance of rating continues to be good as in line with the results shown in Figure 6.
 - ii. We will meet our goal of adding additional rating scales. With the onboarding of a new programmer, and the focus on finalizing program structure and robustness, we have not yet added additional starch rating scales. We will add some MSU scales that are not yet integrated. We will ask industry stakeholders to prioritize which scales they prefer. We will focus on varieties that are relative to Washington State.
 3. *Build a data set to train and test the software.*
 - a. ***Previously stated objectives for 2025:***
 - i. We will continue to collect new tray images internally and with partners to improve training in image recognition and segmentation
 - ii. We will continue to update our trained model files that are publicly available.
 - b. ***What was accomplished***
 - i. We continued to collect images: As already stated in section 2.b.i above, we have been collecting images in 2025
 - ii. Updating of models: The deep learning model of Granny is used to segment fruit into individual images from an image of fruit on a tray. The work we have done in 2024 to build the model for image segmentation has performed extremely well in identifying fruit, and recent changes in Granny allow end-users greater control over how the model removes incorrect calls (happens rarely). Thus, we have not had a need to release an updated model during the 2025 season.
 4. *Provide outreach and training*
 - a. ***Previously stated objectives for 2025:***
 - i. We will update the User's Guide to support use on a Windows PC.
 - ii. We will work with partners to ensure online documentation meets their needs and expectations.

- iii. We will support partners as they test Granny and work with us to improve the software.
- b. ***What was accomplished***
- i. The User's Guide has been updated. As previously stated, we shifted from providing a PC based version of Granny to working with the Staton team to develop an online web-accessible version of Granny. We did not update the documentation for Windows support. However, we did add to the documentation a new section titled "Developer's Guide". This section walks developers (stakeholders with on-staff programmers) through the steps of adding their own analysis modules. Before the end of the project we will also be adding documentation for how stakeholders can write IO (Input/Output) modules such that they can integrate results from Granny into their own in-house databases.
 - ii. We have supported partners. We have checked-in with our industry partners during the year. Because the software is open-source and easily obtainable, several of our partners have downloaded and tested Granny within their own assessment environments. We have received feedback both regarding the functionality of Granny, as well as the GUI that was developed by the Staton lab. This feedback has allowed us to make some important changes they have requested such as the ability to change all analysis parameters, the ability to integrate, improvements to the data outputs. We are currently focused on reading hand-written or typed IDs on each try as requested.
5. Additional Accomplishments
- a. WTFRC project cross integration:
 - i. *Integration with WTFRC project AP-22-101 "Towards next-generation maturity indices: apple biomarker discovery".* During the summer of 2025, our student worker participated in a USDA funded Research and Extension Experience for Undergraduates (REEU). The goal of REEUs is to give undergraduate students exposure to research projects in an effort to encourage new scientists moving into graduate programs and careers in science. The student used Granny ratings to see if more granular and precise granny ratings would improve the machine learning (ML) models that are used to predict fruit maturity markers using gene expression data. Starch ratings from Granny were used as the expected values for predictions. Results showed that Granny performed as good as a single highly trained rater. Next steps are to test if Granny ratings perform better than a variety of human raters with varying levels of expertise. The hypothesis is that Granny will perform better.
 - ii. *Data-Driven Rating Scale for WA 64.* We have begun working with the WTFRC on a project to integrate Granny in a new WA 64 starch scale. Granny ratings show that human ratings are not linear. This integration of effort will hopefully yield new scales that are easier for humans to use.
 - b. Upgrading other rating modules: The objective for this project was to provide a starch rating module, and while this has been our primary focus, we also upgraded the other 3 additional rating modules that were part of Granny before this project, but which were not fully integrated into the new modular framework that this project supported. These include rating modules for percent superficial scald, pear color and blush in pears. The MVC framework we used to redesign Granny from its original state allowed for easy modular integration of these additional rating modules.

Activities for 2026 (If NCE): N/A

Methods

Development of the Granny Software

Granny began as a collection of efforts between the Loren Honaas (USDA), Stephen Ficklin (WSU), Carolina Torres (WSU), and Rachael Leisso (USDA) research programs. When we discovered we were collectively working on the development of software for rating different pome fruit quality traits using image processing and machine learning, we decided to combine our efforts into a single software package we called Granny (a nod to the Granny Smith apples we were using to rate superficial scald). A figure showing the workflow for how fruit rating occurs in Granny is shown in **Figure 1**. Our collective efforts yielded a set of software code, written by each of our different groups that included Python, Java, and ImageJ code. They were sandwiched together to form a single Granny software package. Recently, we published a manuscript (listed in the significant findings section) in the *Plant Direct* journal that describes this version of the Granny software. We referred to this as version 0.5 alpha (alpha meaning it is ready for testing but not production use).

While the 0.5 alpha version of Granny works, it was not optimal. The software needed a major rewrite to use a single programming language, and it needed a proper software design. Additionally, one of the software libraries we were using to identify fruit on a tray stopped being supported. Hence, the software needed a newer package for image recognition. Additionally, we needed to ensure that the software was delivering a high-quality robust starch rating module that could be used for industry applications and for research. Hence our request for this funded project. To meet the software development goals of the project we actively worked on each of these four specific areas:

1. conversion of the software to use a single unified programming language;
2. a complete redesign of the underlying infrastructure to support flexibility and ease of maintenance;
3. upgrade of the segmentation module of the software;
4. and development of a robust starch rating module.

First, we met the goal of consolidating the software into a single programming language. The current version of Granny is now written only in the Python programming language. Python was chosen because it is one of the most commonly used languages for machine learning and AI. Python also has ample image processing and data science modules available for programmers to use within their programs—that makes our job much easier. Python is also extremely popular with data scientists, software developers, and web developers, so it is an attractive language for finding competent programmers who want to learn and use it.

Second, we took the time to completely redesign the software. Version 0.5 alpha was simply a collection of cobbled-together programs. In contrast, our most recent version of Granny, version 1.0 alpha, is now designed in a modern Model-Viewer-Controller (MVC) style framework. This type of framework will support long-term flexibility because it fully separates the software into a plug-and-play type design where modules form the parts of the program that are responsible for the user interface (viewer), the way the software behaves (controller), and the way results are stored (model). This design will allow us to easily add new fruit rating modules, add new user interfaces, or change the way results are stored (e.g., to an in-house database) more easily. This redesign is quite far along, as we have already designed:

- the "viewers" (or interface) for interacting with Granny on the computer terminal (a text-based interface—not yet a graphical interface);
- the "controllers" (or analytical modules) for the rating of fruit that include modules for rating apple starch clearing, pear color, pear blush, and percent superficial scald;
- and the "models" to internally represent results and store them in files on the computer that can be easily opened with a text editor or Excel.

The code for Granny can be found online in an open repository at <https://github.com/SystemsGenetics/granny> (see **Figure 2**). To manage the Granny source code we use a tool called Git (<https://git-scm.com/>) and an online repository called GitHub (<https://github.com/>), which allows multiple programmers to work on the same software code without stepping on each other's toes. It also allows anyone who has programming experience to see how we have designed the code, and contribute back their own changes for review if they would like to be involved in our open-source software development.

One additional long-term benefit of our MVC design is that we now have an Application Programming Interface (API). An API is a set of written instructions and a set of programming "hooks" that allows anyone with programming skills to write their own plugins! We have written such an API. The API documentation is currently found online in the "Developer's Guide" at https://granny.readthedocs.io/en/dev/dev_guide/index.html. The objective here is that because the software is open-source anyone can add to it, including our industry partners who may have sufficient in-house computing experience to do so. This will allow for custom proprietary modules if needed.

Third, we have completed upgrading the segmentation module of Granny. This is the module that identifies fruit on a tray. It first identifies fruit and then "segments" the tray image into individual fruit images. This module requires the use of AI. We previously used the Mask R-CNN package to identify fruits and fruit cross sections on a tray. The Mask R-CNN package is an AI tool that must be trained to recognize objects. It learns to recognize fruit by repeatedly giving it images of the fruit and correcting it when it makes mistakes. Over time, it learns to properly identify fruit. This method was used in the 0.5 alpha version of Granny. However, the package has not been updated and newer versions of Python do not properly support it. Instead, we switched to using the You Only Look Once (YOLO) package which is actively supported and provides the same functionality. Converting to YOLO took some time, but the current version of Granny now fully supports the identification of apples, apple cross-sections, and pears using this tool. The trained models are freely available on the Open Science Framework (OSF) repository for download (and automatically retrieved by Granny) here: <https://osf.io/4jxy5/>.

Fourth, we have upgraded the starch rating module of Granny to use the new infrastructure just described. The results of the rating module are shown in the results section below. The starch rating module currently rates starch by providing a percentage of staining on the apple cross-section. This is different from the starch card ratings we are typically used to using which are subjectively determined using the human eye. However, to help understand how Granny ratings compare to the starch cards, the Granny results are converted to an approximate starch-card rating that is closest to each of the starch rating cards. While we were not expected to upgrade the other rating modules, we did also upgrade the ratings for percent superficial scald in apples, pear color and blush in pears.

During the second year of the project we worked to improve the robustness of the software, to ensure it can be used in a production setting. This involved the development of functional testing. Essentially, functional testing is a separate program that we write that is only designed to stress-test Granny. It provides valid and invalid input and makes sure that every aspect of the software, including the internal parts that the user does not see, responds as expected. This is extremely

important because programmers make mistakes and sometimes in the process of adding a new feature or fixing a reported bug they inadvertently break something else. Functional testing is performed every time a programmer adds new code to Granny and it checks to make sure nothing breaks and results will always be consistent. We needed such testing to make sure that the starch ratings are robust across all of the versions of Granny that are released. Granny now includes a suite of functional testing, with more forthcoming. Writing of functional tests is an ongoing effort and is never completed, but we have a very strong base that ensures core robustness in the software.

Towards Robust Starch Ratings

In order to further evaluate the accuracy of the starch-rating function of Granny, we have worked with partners to generate a testing dataset that consists of images of real-world samples that have accompanying physiological data - namely starch ratings by human technicians. In the first year of the project, we processed the images with Granny to produce digital starch ratings and then compare the results via linear regression (i.e., granny vs. humans). We analyzed 245 images of trays of 'Envy' apples and 59 images of trays of 'Honeycrisp' apples that were provided by Austin Wilson (Allan Brothers Inc.). There were typically 25 apples per tray in each image. Starch ratings of 'Envy' apples ($n=4825$) were classified according to the Enza starch scale, and ratings of 'Honeycrisp' apples were classified according to the 'Honeycrisp' starch scale ($n=750$). Both the human ratings and granny ratings of the fruit were averaged across the tray. The tray averages were subject to linear regression analysis, Pearson's R was calculated.

Outreach and Training

To ensure that Granny is accessible to anyone, we have created a user-friendly User's Guide that is freely available online. This User's Guide walks users step-by-step through the process of rating fruit using Granny. We also provide example images that a user can use to test Granny before experimenting with their own images. We use a popular Read the Docs format for creating these documents. One benefit of this is that the documentation is freely available online, and anyone can provide suggestions for changes if they see a typo or want to suggest clearer language. The User's Guide can be found here: <https://granny.readthedocs.io/> (see **Figure 3**).

As part of our outreach efforts, we have worked with our industry partners to develop an imaging station design. The design instructions for the imaging station are available in the online User's Guide. We have also created a template mat used to organize items needed by Granny. The mat provides instructions for where to place the tray, where to place a color card for calibration of the image processing, and a place for sample information notes. An image of the template within an image station can be seen in **Figure 4**. An example of the template mat being used within an imaging station can be seen in the demo image of the User's Guide screenshot of **Figure 3**.

Results and Discussion

Granny accurately rates starch clearing

In the Nguyen et al. 2024 manuscript cited in the "Significant findings/results from 2024" section, we provided an initial analysis of the starch rating module using 38 fruit images. To demonstrate further the analysis of Granny's starch rating function using the newly rewritten Granny v1.0 alpha, we performed a more in-depth analysis using the 245 'Envy' apple tray images and 59 'Honeycrisp' apple tray images that we obtained from our partners. The starch rating analysis in Granny consists of 3 key steps: fruit identification, image segmentation, and calculation of the percent starch clearing in each fruit sub-image (see **Figure 1 panel D2**). Granny provides a rating of 0 to 1 indicating the percentage of starch clearing on each cross-section of the apple, where 0 indicates no starch clearing and 1 indicates complete clearing. For each tray, an average starch clearing percentage is calculated. Lastly,

Granny converts its rating into a starch-card value appropriate for several commonly used starch cards. In this case, the Envy and Honeycrisp starch cards were used. This conversion is provided to help users better conceptualize the rating and to allow us to perform comparisons. We expect that the Granny rating from 0 to 1 is more accurate than the converted rating. The main goal of our tests was to see if Granny produced similar starch ratings to humans in a real-world setting. The results for ‘Envy’ apple trays are plotted in **Figure 5**. Results showed a high degree of correlation between human ratings and the ratings provided by Granny ($R^2=0.64$). The analysis of ‘Honeycrisp’ apples (not shown) exhibited a similar result ($R^2=0.61$). These tests showed that Granny is accurate and that Granny seems to slightly overestimate the degree of starch clearing when compared to human ratings. Because there is a strong correlation between human ratings and Granny ratings, future calibration of the software will be straightforward.

USDA AI Innovation Fund Award to support the development of user interface

During the project period, we received additional support from USDA to develop a graphical user interface (GUI) for Granny. In lieu of creating a Windows interface (as was an objective for 2025), we felt an online interface that would make Granny accessible in the lab or the field would provide greater flexibility. Briefly, along with new collaborator Dr. Meg Staton from the University of Tennessee at Knoxville, we developed a simple user interface for the starch rating function of Granny. Her team are experts in development of citizen science mobile and web-based applications. The current version of Granny uses a command line interface (commands given to the software in text form), and this graphical version will always remain available. The command-line version is especially useful in research applications. However, the new GUI is now available for testing, has been demonstrated to industry partners in August of 2025, with requests for feedback, and makes Granny easier to use in a commercial setting. Our design concept for the GUI is shown in **Figure 6**.

Figures

Figure 1. Granny key functions, adapted from Nguyen et al 2024. Granny performs four steps to process an image and provide a rating. (a) First, images of fruit trays are provided as input (apple or pear). (b) Second, individual fruits are detected on the tray. The figure shows individual fruits colored differently to show how they are identified. (c) Third, each fruit is segmented into individual fruit images. (d) Fourth, ratings are performed including detection and scoring of D1) superficial scald, D2) starch content, D3) pear peel color, and D4) pear blush percentage.

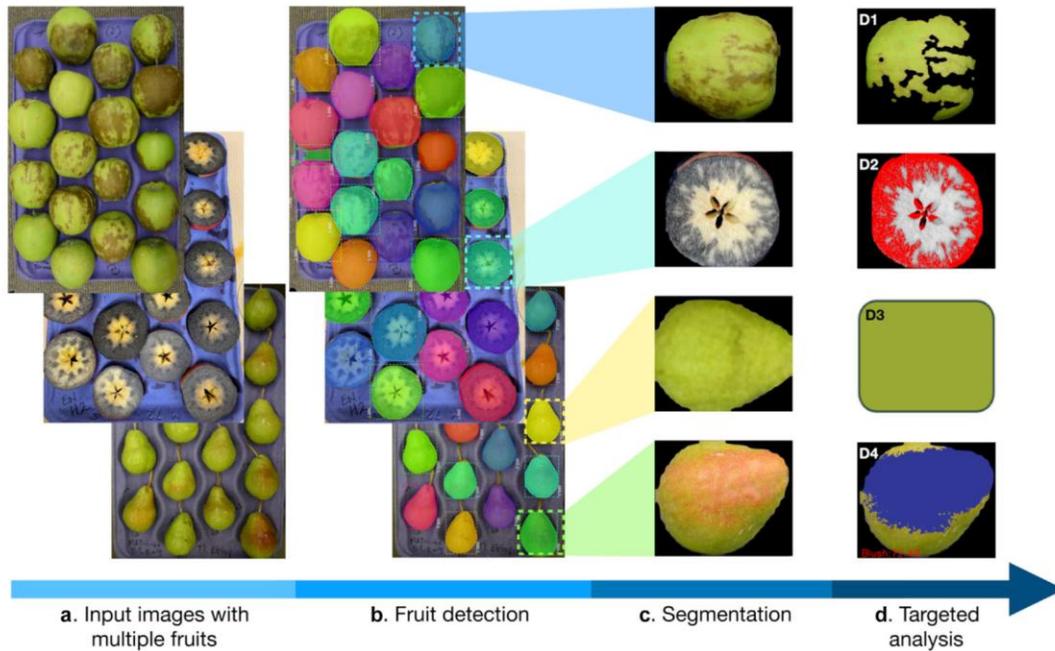


Figure 2. Granny Source Code Repository. The following is a screenshot of the Granny software source code repository on GitHub at <https://github.com/SystemsGenetics/granny>.

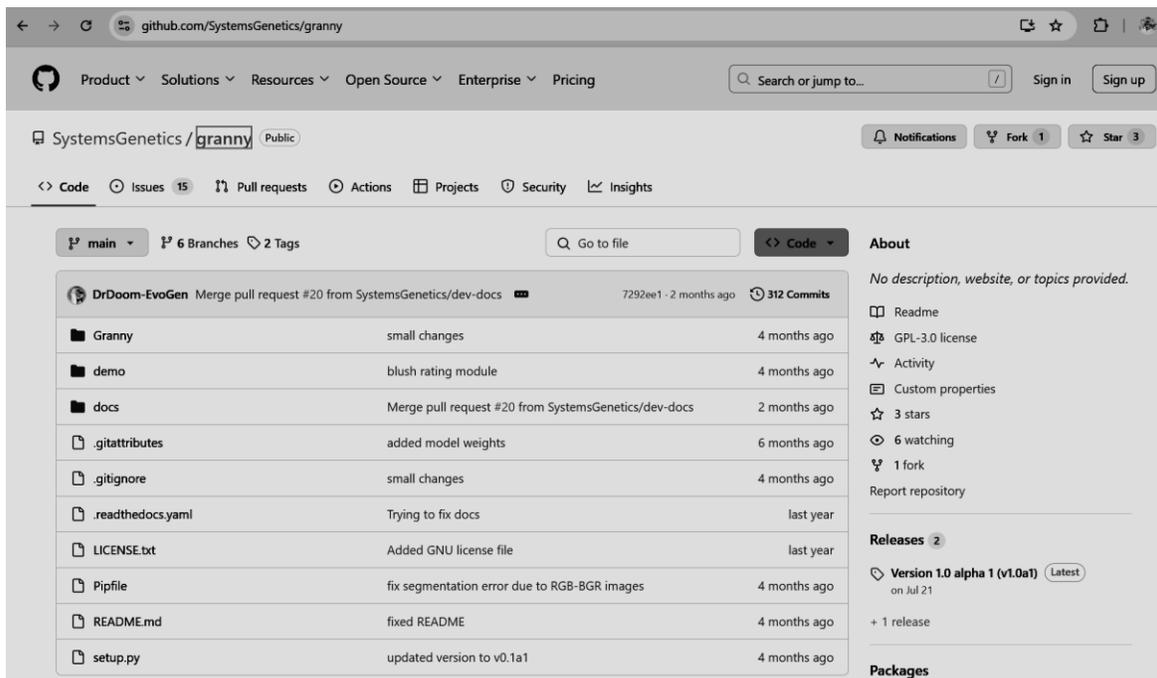


Figure 3. A screenshot of the Granny User's Guide showing the instruction page for rating starch. Available at <https://granny.readthedocs.io/>.

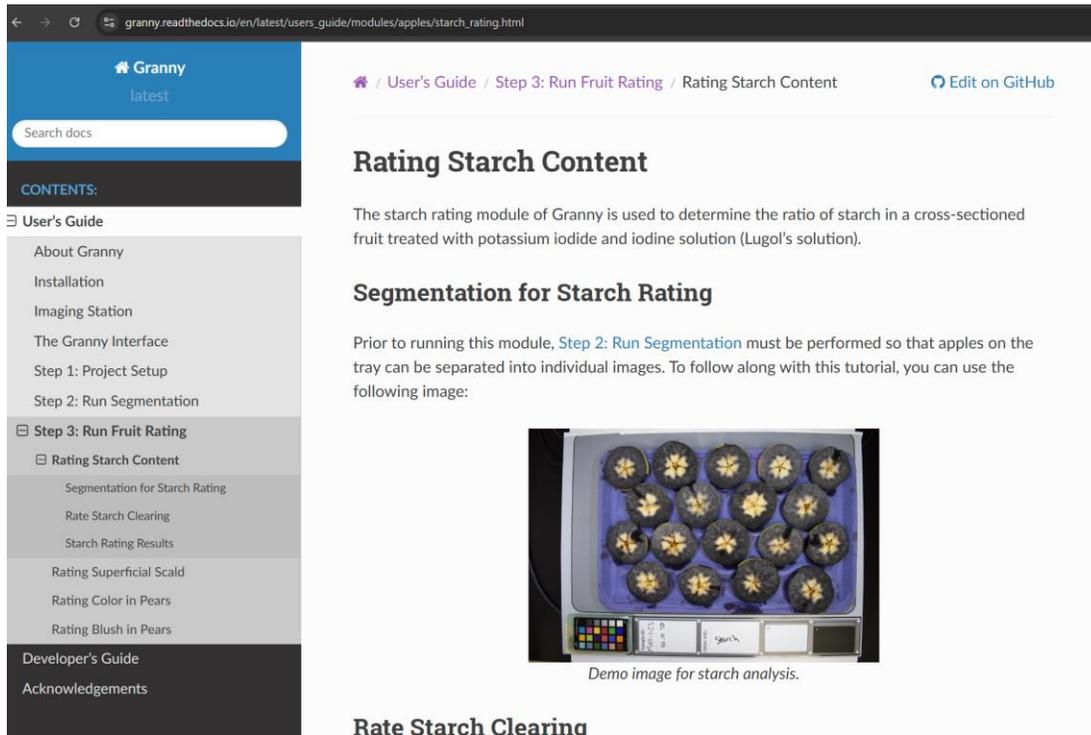


Figure 4. The Granny Imaging Template Mat. The design for the imaging station and the template mat are described in the Granny User's Guide. The image shows the template mat which indicates where the tray should be positioned, where a color card should be positioned, and an area for sample information.

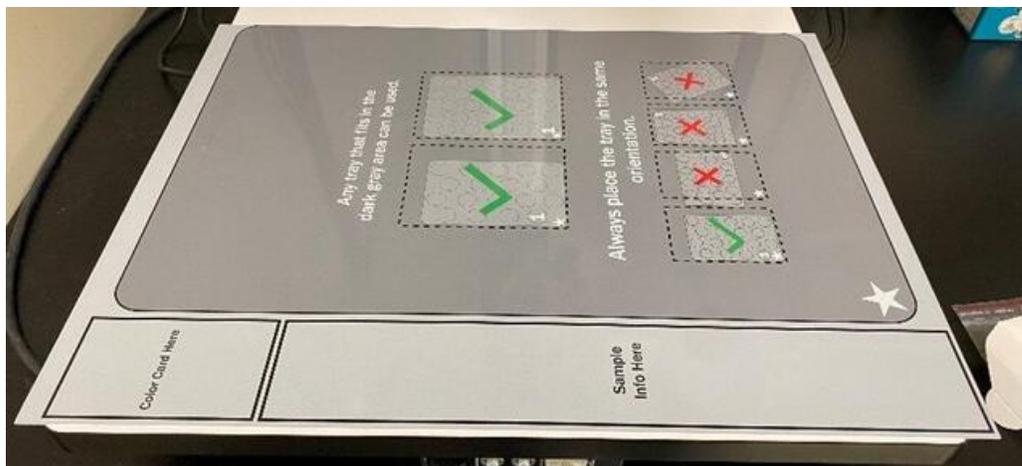


Figure 5. Analysis of commercial samples shows high accuracy. Average starch ratings of 245 Envy apple tray images are plotted in the full figure where human technician ratings are represented with circles and Granny's ratings of the same trays are shown as triangles. The 245 tray images are ordered on the x-axis using the percentage of the cross-section not stained by starch that Granny produced (the lowest on the left). For comparison, the actual

value plotted is the Granny rating converted to a starch card value. A strong linear correlation is observed between human and Granny's ratings as is shown in the plot on the lower right corner.

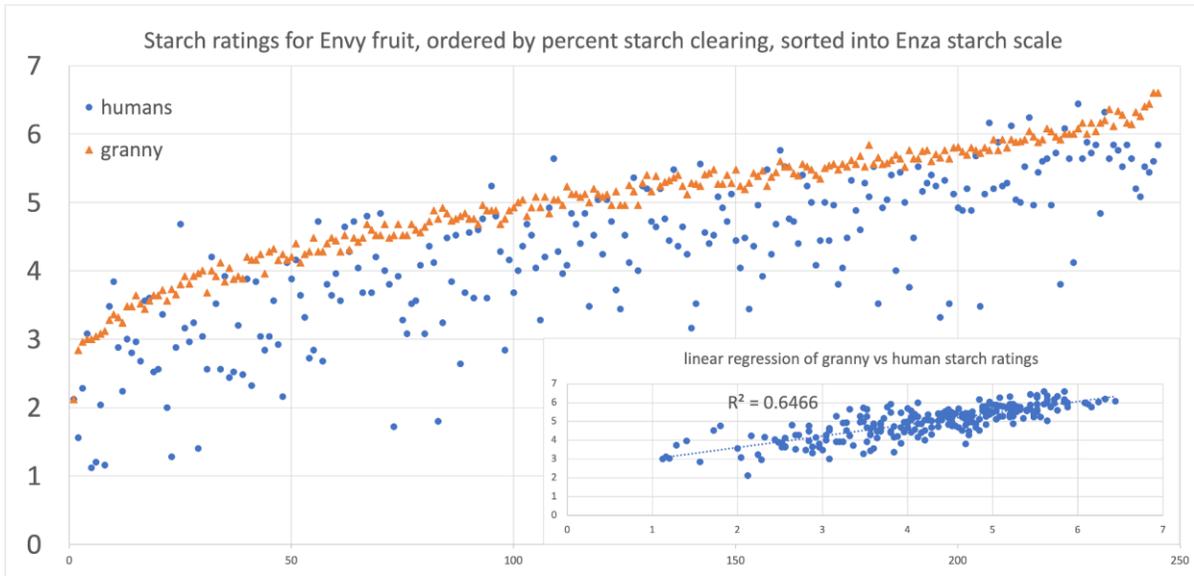
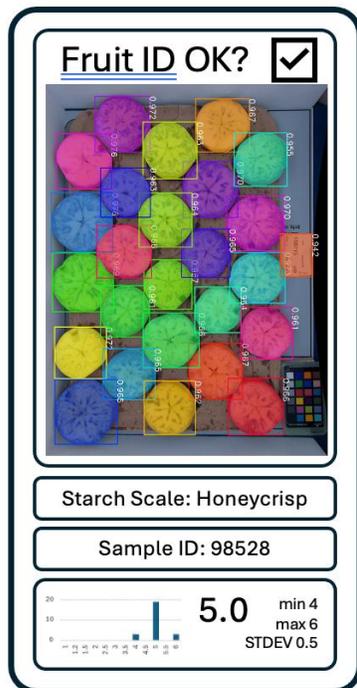


Figure 6. User interface concept. We have designed a graphic user interface (GUI) with Dr. Meg Staton's research group. Once an image is uploaded and processed by the GUI, users can review the image and the fruit detection results, verify the Samples ID, select the reference starch scale, and the software produces a summary graph and tray-level statistics.



Executive Summary

Project title: *Granny – Image-based Analysis of Fruit Quality; TR-24-100A*

Key words: image-based phenotyping, starch rating, fruit maturity, machine learning, fruit quality

Abstract

This project delivered an important advancement in automated image-based fruit quality assessment software. *Granny*, an open-source platform originally developed to rate superficial scald, has now been fully re-engineered into a modern, modular system capable of robust and reproducible apple starch clearing evaluation. Across the two-year period, the project team completed a full rewrite of the pre-existing disjoint software into a single unified Python package, implemented a modular expandable software framework, and integrated a modern deep-learning model for identification and segmentation of individual fruit on a tray. The primary objective of this project was to deliver a starch-rating module that provides granular and consistent ratings, avoiding the pitfalls of human rater bias. Additional rating modules for superficial scald, pear color, and pear blush were also upgraded and integrated. Extensive functional testing ensures reliability across future versions. Image-based starch ratings produced by *Granny* correlate strongly with human rater assessments across commercial samples of ‘Envy’ and ‘Honeycrisp’ fruit. Its design allows industry users and researchers to expand its capabilities by integrating custom analysis modules and database systems.

Industry collaboration was central to success. Growers and packers provided images for training and validation, and several partners tested the software in their internal workflows. User documentation and developer guides were substantially enhanced, and design support was provided for imaging stations and tray-layout templates. A new graphical web interface—supported by a supplemental USDA award and developed in collaboration with the University of Tennessee—extends access to Windows, macOS, and field-use environments without requiring installation.

This work provides to researchers and the industry, objective, standardized, and scalable fruit maturity measures and provides a flexible platform for additional trait-rating modules.

Leveraged Funding Summary

Over the life of the project, development was supported by funding from:

- All funds from the Washington Tree Fruit Research Commission for this proposal.
- Additional funds leveraged to support the development of *Granny*:
 - USDA ARS AI Innovation Fund to support creation of a graphical user interface for *Granny*.
 - Data storage equipment from the National Science Foundation (NSF) purchased for a previous project that ended prior to this award was used to support development and testing of *Granny*.
 - Funds from a USDA Research and Extension Experience for Undergraduate (REEU) supported purchase of a high-performance workstation for the student programmer.

Combined leveraged support exceeded \$400,000, providing computing infrastructure, programming support, and user-interface development to enhance outcomes beyond the base award.

