Sprint 4 Report
Product Name: Automatic Cell Counting
Team Name: Cell Counting Project
Date: 06/01/2021

Actions to stop doing:

- Stop accruing/ignoring Architectural Debt

Actions to start doing:

- Using Trello to organize our tasks. We did have a very good organization on Trello during the times we did use it. It helped us keep the whole sprint in perspective and keep us from getting lost in the weeds. But we ignored it too much this sprint and it was not as useful.
- Integrate the model more thoroughly. We had used too much repurposed code for the purposes of making sure we weren't breaking something that we didn't understand.
- We need to test our helper scripts more thoroughly.

Actions to keep doing:

- We were testing in Google Collab, so we were making all of our changes in the Google Collab. We commit our changes from Google Collab to our GitHub more frequently.
- We kept track of administrative tasks (things that didn't fit on the scrum board) separately to make sure we got it all done

Old actions to keep doing

- Daily scrum meetings are within the time limit. We were good at summarizing our progress and needs while getting all necessary information across.
- Important documents and other notes are in the Drive for everyone to use. It's well organized and there has been very little confusion as to where documents are located.
- Keep code in separate branches and merge regularly to main.
- Working together to solve problems we encountered and helping each other out.
- Team communication is really good and we are aware of what each of us is working on and whether we need help or not.
- Ask TA for advice more often. Akila has a ton of really useful prior knowledge about these kinds of projects.
- Keep updating the burnup chart regularly.
- Keep communicating in Discord all our problems and keep everyone in the group informed.
- Keep revising user stories when progress is not as expected.

Work completed:

- We can download images and save their counts to any location the user wants
- We can open a folder to predict on multiple images sequentially
- We refactored our code and added comments
- We improved the pathing so that it can run from other locations

- We made a README to guide the user through executing our app
- We changed the directory structure and removed unnecessary files

Old work completed:

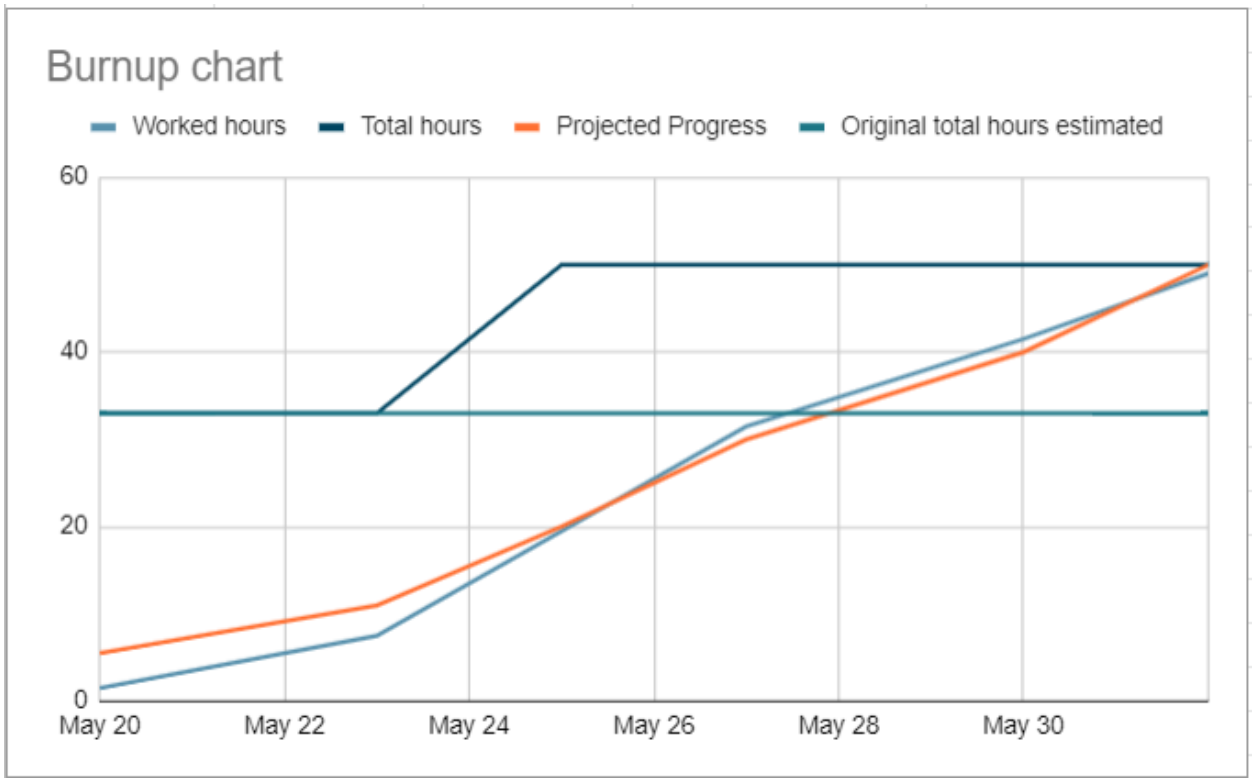
- We completed the UI download option
- We are able to train the model without running into input errors due to the difference between our Jsons and their jsons.
- We are able to run predictions without encountering formatting errors.
- We finished annotating images
- We have trained two models (nucleus and balloons) that can predict on cell images that are similar to the ones we trained with!
- We have a python notebook that we can use to quickly download our code and start working on it.
- We have a script to convert the jsons to the correct format for the model.
- We have a way of creating dummy masks to run predictions
- We have designed a more ideal architecture for our codebase
- We rebased our code to be more modular and abstracted
- We have made a test fixture for our UI and have written 3 unit tests for our code.

Work not completed:

- We want to train with more of our data to have a larger variety of cell image types that we can accurately predict on.
- We want to add some more unit and integration tests.
- We want to be able to tell users the average size of cells in the images.

Work completion rate:

User stories completed: 5
Total number of estimated ideal work hours: 49
Total number of days in past sprint: 14
User Stories/Day: 5/14 = 0.36
Ideal Work hours/Day: 49/14 = 3.5


Burnup chart
Worked hours
Total hours
Projected Progress
Original total hours estimated
60
40
20
0
May 20
May 22
May 24
May 26
May 28
May 30