

Team Member Reintroduction and Restating of Roles

From our Roles & Responsibilities Documentation...

Avery Cameron - Lead Machine Learning Technician

- Lead development of classification model and documentation
- Understand how to integrate with and operate within Amazon SageMaker and other products
- Manage our GitHub structure and CI/CD
- Record meeting minutes

Raymond Knorr - Lead UI/API Developer

- Lead API and Front-End Developer
- Manage integration and documentation of our software with Prairie Robotics
- Manage Trello board for workflow organization
- Manage communications between URStreamSight & PrairieRobotics

Noah Rowbotham - Project Management/Flexible Developer

- Organize project meetings and milestones
- Organize meetings between URStreamSight and mentors
- Manage project documentation

Although we have defined roles, we intend to provide insight and support to each other regarding all other aspects of development during this project.

What is URStreamSight?

We intend to produce a software solution that will monitor the quality of municipal recycling and give meaningful feedback to the municipalities. With this information, municipalities can deliver targeted education to reduce contamination in recycling streams and increase the quality of recycling.

Progress Thus Far

Late September - Project documentation, refining project goals, and scheduling milestones

Prior to our first Scrum meeting we began developing documentation for our project. We used this time to take our fairly loose ideas from brainstorming with Prairie Robotics and make them more concrete. We defined clear goals, scheduled milestones, and organized meetings. From this process we started to identify key unknowns that we needed to clarify such as:

- 1) Problem complexity, is it possible to develop a classification model that performs well with a high amount of recycling and obfuscation? Since a large amount of data to classify is state of the art technology.
- 2) *StreamSight API Purpose: What is all handled by the API?* At this point we believed that the API would need functionality to manage uploading images to the Amazon cloud services, filter out low-quality images, and provide data to our front-end UI.

October 13th - Discussed classifying models with Dr. Yow, updated our expectations

Our initial goals were to create a model that could classify all items in a households recycling bin as recyclable or not. That way we could then compute a score based on the amount of non-recyclables present in their bin. Based on our early research, we were concerned that this problem may be too complex for us to solve due to the amount of items and high levels of obfuscation in a single frame.

Now, we want to develop a model that can only identify common non-recyclable items that frequently cause recycling to be too contaminated for processing. If we can identify items such as bags of trash, propane tanks, lithium ion batteries, aluminum cans, glass bottles etc. we can still develop a method of rating a household's recycling based on the number of these specific items.

October 14th - Refined our project scope following a meeting with Prairie Robotics

To further define the scope of our project and ensure we are meeting our clients needs we recently had a follow-up meeting with Prairie Robotics. What we learned from them is that the technology needed to take images of recycling and upload those images to Amazon cloud storage is handled on their end. This significantly modifies the tasks that our API needs to manage.

We also thought we would be developing a UI to meet the needs of individuals who recycle. But, based on Prairie Robotics' insight, households that recycle poorly likely do so because they haven't sought out the resources to learn and improve. Any software we develop is likely to not be found as the present resources are. Because of this, it is likely far more beneficial if we create a UI that meets the needs of the municipalities who use our software. To meet their needs our UI will display various screens such as a heatmap of the municipality that visually shows trouble areas for recycling or tabular data that lists the top 10 or 20 worst households for recycling. With this information municipalities can develop targeted education or reprimands for these identified problem areas.

In summary, URStreamSight is software that:

- 1) Labels poor-quality images as bad data or training data to ensure our classification model only trains on good-quality images
- 2) Uses a computer vision classification model to identify and classify high-risk non-recyclables present in a households waste bin
- 3) Has a back-end API that runs the image filtering job, and serves the requests of our front-end UI for fetching output data from the classification model
- 4) Has a webpage that provides meaningful representations of the data to municipalities for identifying problem households

The Next Steps

- 1) Meet with our mentor Dr. El-Darieby and get more insight regarding the design and implementation of our project (October 20th)
- 2) Finish hifi prototypes (October 26th)
- 3) Begin designing our API and learning how to integrate with Amazon services (October 26th)
- 4) Begin using current open source computer vision models to classify recyclables, use this research to inform our future decisions (October 30th)
- 5) Finish API design documentation and begin or complete front-end designs (Mid November)
- 6) Deliver first MVP by end of term that demonstrates the API and front-end with mock data

Show Lofi prototypes

Team reflection

Do we feel "on track"?

Yes, we have put a lot of effort into planning and documentation (ideation) so that when we begin our designs we have a strong understanding of what we are creating.

What progress does the team particularly feel good (great) about?

We feel great that we essentially redesigned our project over the last three weeks to meet the requirements of the new information we gained. We have avoided a lot of potential stress and wasted development time by working hard to clarify scope and goals.

What barriers (if any) does the team feel is a current impediment to success?

We currently feel barrier free, but we are nervous for next term as data collection and model development could present significant challenges or barriers to us. We are actively always trying to mediate these future risks as we proceed with development.

What help (if any) does the team require to move positively forward?

We would like to get feedback from our mentor to hopefully find more problems to address. That way we ensure our designs are not plagued with too many uninformed poor decisions. We also need to organize access to Prairie Robotics' Amazon services and GitHub so that we can include integration as part of our design.

What questions or concerns does the team have (if any)?

Is there an issue with integrating with the existing front-end and api of PrairieRobotics? How does this work for proprietary purposes?