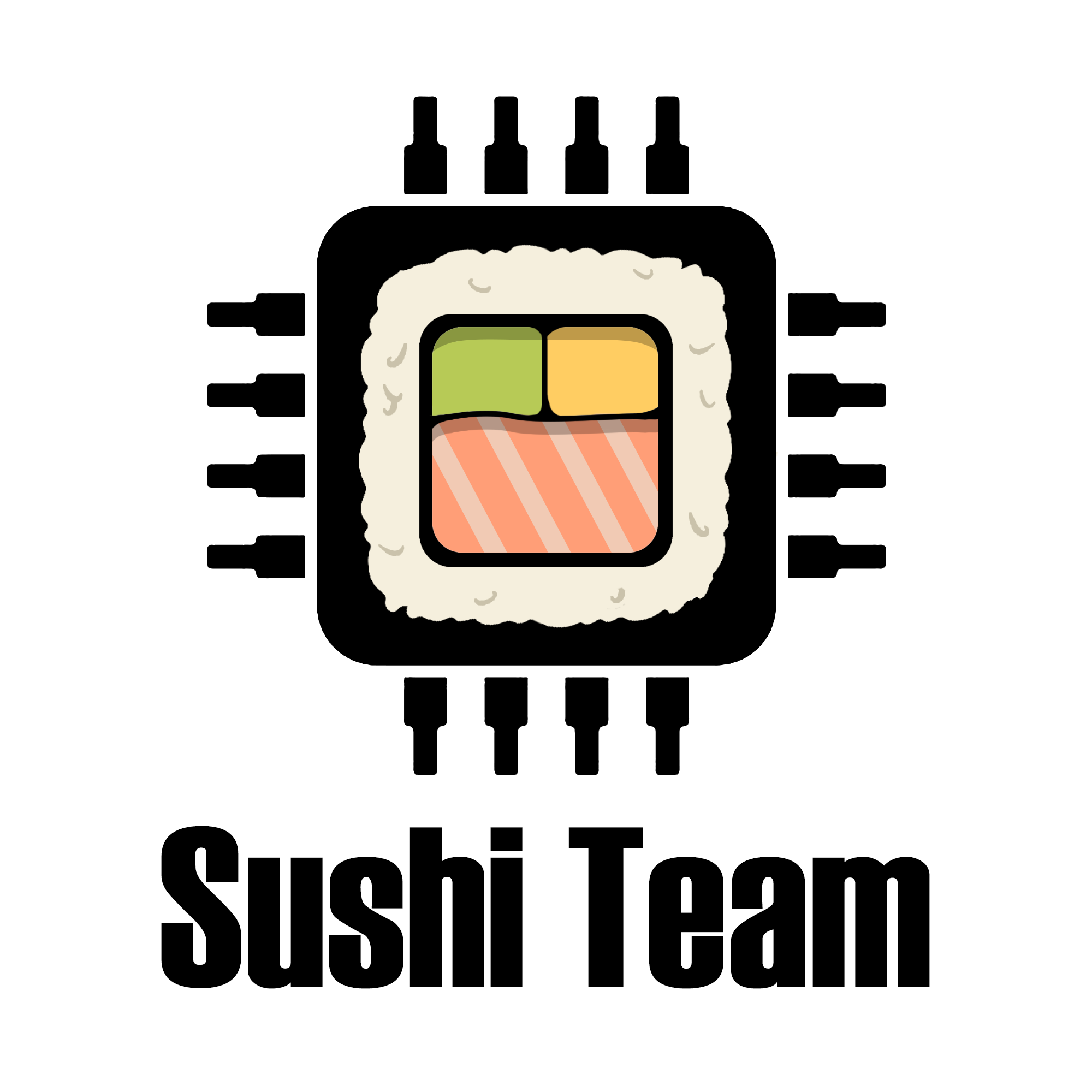
**Final Presentation Meeting Agenda**

When & Where: May 17th, 2022 ~ 1:45pm - 3:45pm @ Zoom

Attendees: Dr. Jared Macshane, Dr. Shaunn-inn Wu, Sushi Team



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2:25-Beginning/Formal Greetings (Noah - Team/Scrum Leader)**

**2:26-Purpose (Jordan - Documentation/Training)**

* Introduce the client organization and explain the purpose of the project.

**2:27-M.L Project Requirements / Network Diagram (Jordan - Documentation/Training)**

* Develop a system to receive and process Google Street images from a user (the Web App team).
  + Google Cloud/ML-Server Instance | TCP connection
* Develop a Machine Learning algorithm that must be able to detect instances of litter in the received Google Street images.
  + fixIT Dataset | Roboflow | Converted and Re-annotated Dataset | YOLOv5 ML-Algorithm
* Once the litter instances have been identified, the system must then process and output the results into a suitable format, like JSON, for the user (Web App Team).
  + Processed Google Street image results
* Once the results have been made, the system must then return the processed data to the user (Web App Team).
  + TCP connection
* *Train the final algorithm to categorize different kinds of litter (Contingent on the completion of other requirements)*

**2:30-Project Management (Juan - Project Manager)**

* Recap the project’s timeline, pertaining to both JADs and Prototypes.
  + JADs - Identify & finalize our requirements and technical specifications for the project.
  + Prototypes - Work towards the construction of our system to receive, process, and detect litter in Google Street images from a user, for our Machine Learning project.
* Recap main development tasks, and subtasks, within the timeline of the project.
  + Develop Image Processor
  + Apply Image Processor
  + Results Formatting/Web App Integration
* Recap estimates vs actuals, for both Hours & Costs.
  + Costs based on a $23 hourly wage.
  + Google Cloud server cost $0.37 cents per hour to run.
  + All of our costs come from not only our main development tasks, but also from our project management tasks, meetings, documentation, and server usage.

**2:34-Development Process (Keith - Programmer / Noah - Team/Scrum Leader)**

* Go over our initial dive/research into how to approach our Machine Learning project.
  + TACO/Mask R-CNN.
  + VGG Image Annotator
* Go over our finalized tools & resources that were used for the development of our Machine Learning project.
  + YOLOR - An object detection model
  + Roboflow - A general purpose Machine Learning site
  + Google Cloud - A cloud computing service platform
  + YOLOv5 - Another object detection model
* Our Google Street image Datasets
  + The Old Dataset - From a previous year’s group (fixIT)
    - Single “Litter” Class
  + The New Dataset - Assisted from the Web App team
    - Multiple Classes of Litter
  + The Merged Dataset - A combination of both the Old and New Dataset
    - All classes collapsed into single “Litter” Class
* Our Machine Learning Training with the YOLO models on Google Cloud
  + Precision, Recall, and maP
  + Experience with YOLOR
  + Experience with YOLOv5
    - Choosing over YOLOR
    - Modified the YOLOv5 code
    - Hyperparameter Evolution training
  + Results with YOLOv5
    - Precision, Recall, and maP levels
* Our Machine Learning functionality with its Detection and Scheduling Hook for users.
  + DEFAULT values for certain variables, specifying file paths for both the Source and Output of the detection code.
  + TCP pipeline, from the Client (Web App team) to the Server (ML Algorithm), through Socket Programming.
    - Start sending multiple images/files to Server.
    - Receive and Process images.
    - Save and Send results back to the Client.
    - Client receives results and Saves them.
  + Features of user Detection code.
    - Constantly running
    - Can take over 100 of images
    - Max of 70% confidence level
  + Accommodating for different types of Google Street images.
    - (Low resolution) Street View Static API vs (High resolution) Screenshots

**2:44-Final List of Deliverables (Miguel - Programmer)**

* Provide a functional litter detection algorithm that consists of:
  + Getting an input of Google Street images
  + Processing them through the litter detection
  + Output the data, consisting of images with bounding boxes, and a JSON file, signifying the total amount of litter detected in the picture.
  + Schedule a hook by which the Web App team can use the algorithm.
    - KABML\_server.py
    - KABML\_client.py
  + Send the output data to the Web App.
* To achieve this, we are using the YOLOv5 model as the engine for litter detection.
* NOT achieved - Conditional litter detection feature:
  + Categorize the litter (e.g. Plastic, Paper, Organics,...).

**2:46-Documentation (Miguel - Programmer)**

* Reports and Manuals for new users and programmers.
  + Roboflow: A general purpose Machine Learning site.
  + YOLOv5: An Object Detection Algorithm Model
  + Google Cloud: Cloud Computing Service Platform
  + TCP Connection & Detection Guide
* All located in our GitHub link located below.
  + <https://github.com/SushiTeam2022/KAB-ML>

**2:48-Challenges (Miguel - Programmer / Keith - Programmer)**

* Old Dataset
  + Poor image resolution & quality
    - Stuck with Google Street API images due to consistency.
  + Poor litter annotations
    - Had to re-annotate ourselves, for other sources were incompatible with the Google Street view images we were working with.
* New Dataset
  + Roboflow/Dr. Schultz’s students
    - Students were added to the project and were given specific instructions on how to annotate.
    - However, they did not follow instructions, and progress on the new dataset was negatively affected, affecting both its quality and size.
* The Models
  + General connection challenges
  + MASK R-CNN
    - Needs Masks - Old Dataset was only Bounding Boxes
    - Not updated in years - would need overhaul
  + YOLOR
    - Not routinely maintained
    - Limited to bounding box annotations
    - Modifications to code were needed to work with latest tools and releases
  + YOLOv5
    - Faster image processing results in lower accuracy
    - Constantly updating

**2:52-Future Enhancements (Jordan - Documentation/Training)**

* A better Dataset:
  + Larger-sized images
  + More pixels to define objects
  + Less blur from compression
* A new & more fortified annotation process:
  + Better control over your workspace/Dataset
  + Have a more dependable platform
* More GPU:
  + Speed up the overall process of the project
  + Multiple process/detecting of litter at one time.
* Expand to categorizing litter:
  + Categorize the litter (e.g. Plastic, Paper, Organics,...).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2:55-As a team - communicate with everyone for Q&A.**

**3:00-Meeting adjourned!**