# Programming for Internet of Things

Object classification and IoT Grocery Sync

Alyana Nelson Ramprakash Sridharan

#### Task/Problem

- A kitchen-based system that allows you to effortlessly add items to your grocery list.
- Simply show the food product to the camera, and it will automatically update your 'grocery cart'. such as an Apples, water, yogurt, cereal, oil, etc.



#### Components of the system

#### Hardware Components:

- Raspberry Pi 4B
- Webcam

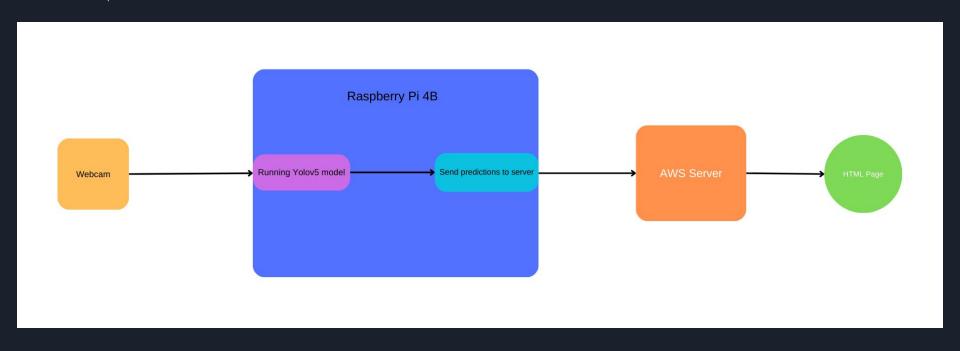
#### Software Components:

- Yolov5 Deep learning model for object tracking and classification.
- Node server.
- Webpage to track the grocery list.





# Interactions



#### Tech stack





- Python
- Pytorch
- Node js
- Javascript, HTML, CSS
- MongoDB



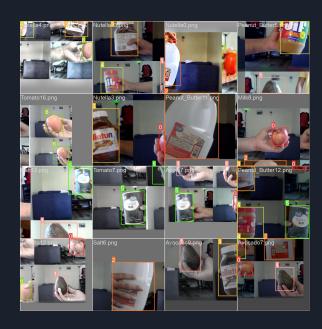




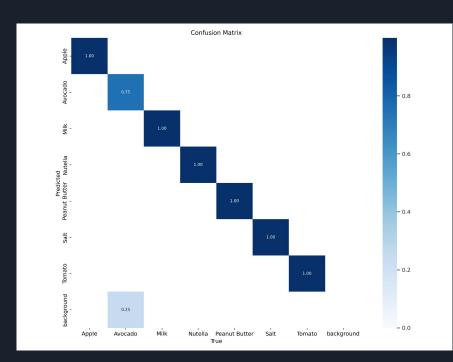
#### Model training

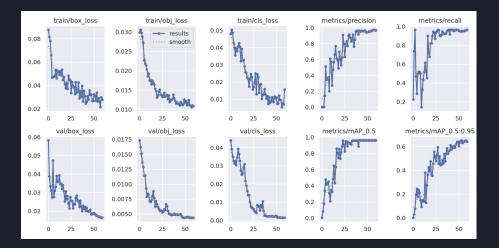
- The training images used for the model are shown below. The images are subjected to morphological operations (shear, zoom, mosaic) etc and fed into the model.
- The dataset consists of 20 images for each class with a total of 140 images.





# Training Result

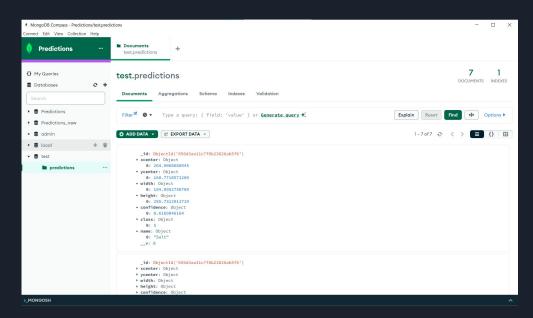




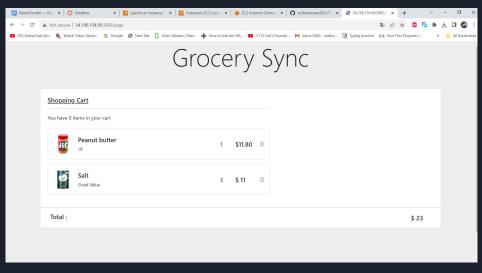
- The model is trained for 50 epochs
- The confusion matrix and the loss graphs are shown above.

# Model deployment

- The trained model is deployed to AWS server and MongoDB is used to handle the data.
- Mongoose library is used to add data to the database.



#### Final Result



```
ubuntu@ip-172-31-27-46:~/Final project IOT/node server$ node node server combined.js
(node:19801) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
listening...
Connected to DB successfully
ubuntu@ip-172-31-27-46:~/Final_project_IOT/node_server$ node node_server_combined.js
(node:19816) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
 (Use 'node --trace-deprecation ...' to show where the warning was created)
listening...
Connected to DB successfully
ubuntu@ip-172-31-27-46:~/Final_project_IOT/node_server$ node node_server_combined.js
(node:19833) [DEF0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
listening...
Connected to DB successfully
ubuntu@ip-172-31-27-46:~/Final project IOT/node server$ nano index combined.html
ubuntu@ip-172-31-27-46:~/Final_project_IOT/node_server$ nano node_server_combined.js.html
ubuntu@ip-172-31-27-46:~/Final project IOT/node server$ nano node server combined.js
ubuntu@ip-172-31-27-46:~/Final project_IOT/node_server$ node node server combined.js
 (node:19868) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
listening...
Connected to DB successfully
ubuntu@ip-172-31-27-46:~/Final project IOT/node server$ node node server.js
(node:19886) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
listening...
```

#### Challenges

• The Deep learning model is not properly optimized for all classes, but good enough detection accuracy for this application.

 The model is not deployed in Raspberry Pi 4 but when deployed would work the same as in the computer.

• The HTML page is hosted as a single file using res.sendFile() from the node server. The js files and style files had to put in the same file to make it work.

• One other way is to use ejs to render the file.

• The database query operations such were hard to perform as a beginner.

# THANK YOU