Ecobin

Charles Thao, Aditya Wikara, Shreenidhi Jayaram,
Hayato Nakamura, Kevin Sinaga

Problem Statement

- The United States generates some 260 million tons of waste in 2018
- About 80% of what Americans throw away is recyclable, yet our recycling rate is only about 30%.

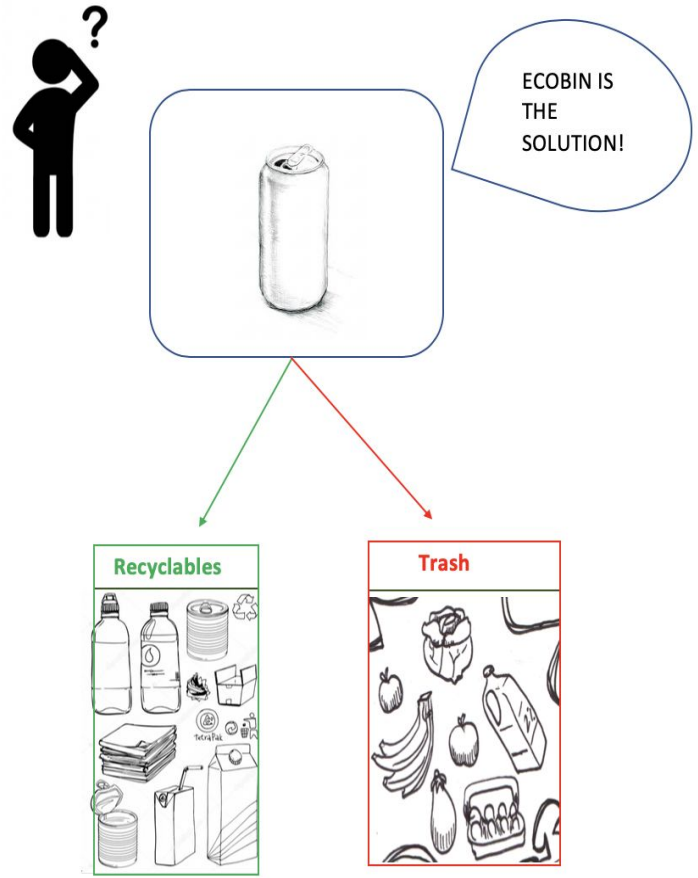
“Recycling takes too much of my time”

“It’s complicated to recycle”

“I don’t know exactly what can be recycled”

Introducing : Ecobin

- An smart IoT waste sorting system
- Increases household recycling by *40%*
- Aimed Accuracy: 75%+

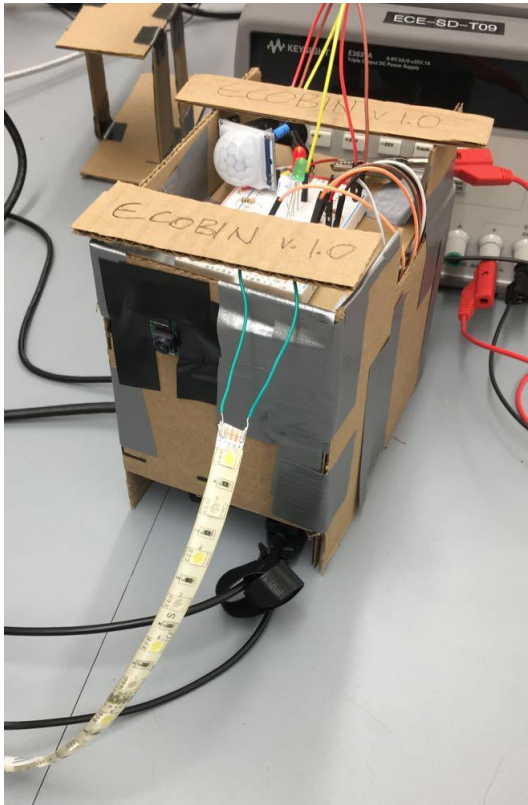


Requirements

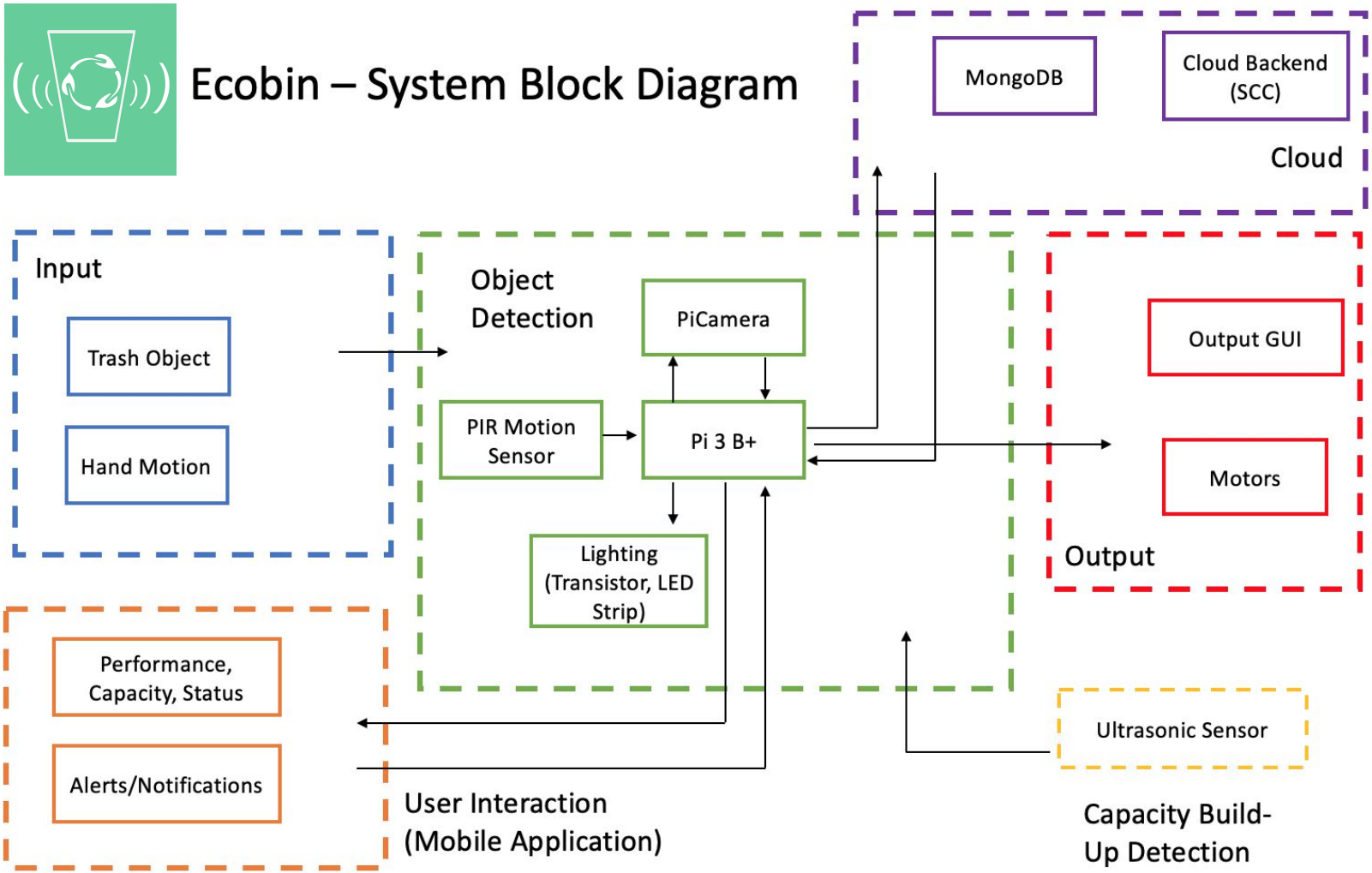
Hardware:	Software:
■ Capture image in desired environment ■ Pre-process image (crop, pixels) ■ Configure all sensors and circuitry (handle all signals) ■ Enable RasPi to communicate with an API, backend, database	■ Correctly detect objects with >70% accuracy ■ Stable cloud back end with 99th percentile latency ■ Interactive GUI with LCD display ■ Stable database ready for multiple users

Recap Before Prototype 2

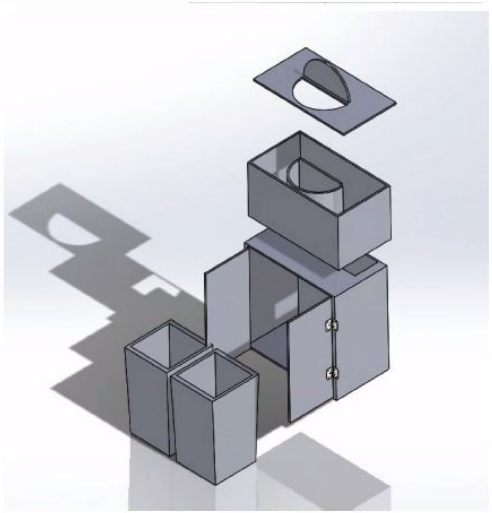
- No testing environment for object detection
- Faulty motion sensor
- Trash build-up detection feature did not exist
- Recognition using VGG16 (pre-trained model)
- No database/API to interface RasPi with backend
- No GUI to output the results
- The main program ecobin.py was not modular

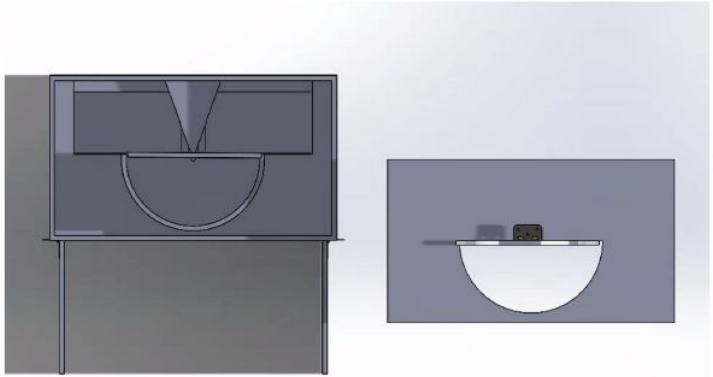


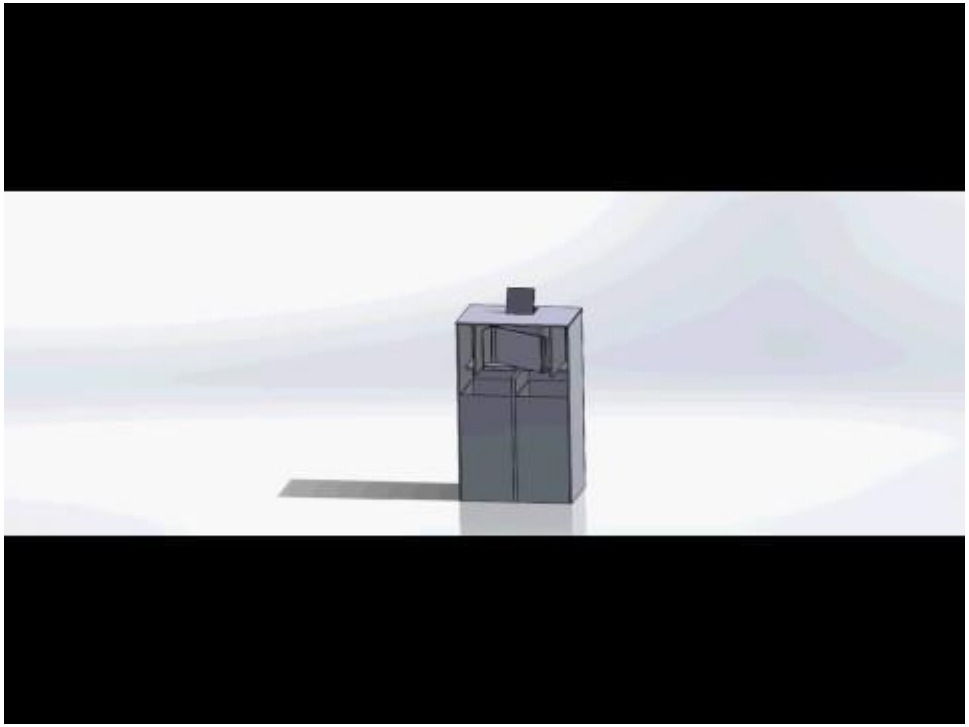
Ecobin – System Block Diagram



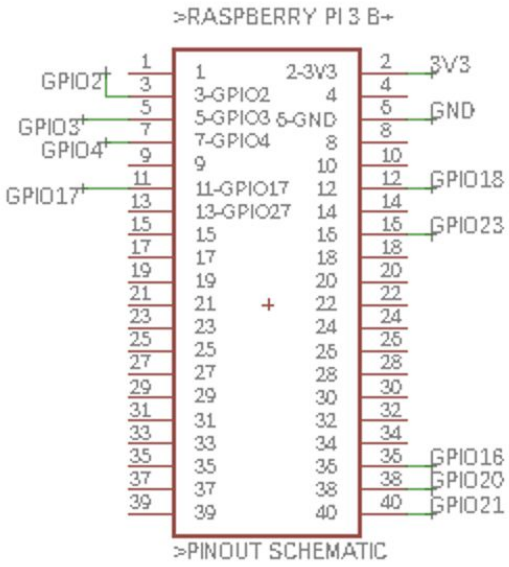
Mechanical Design

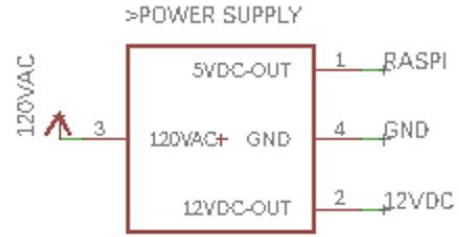


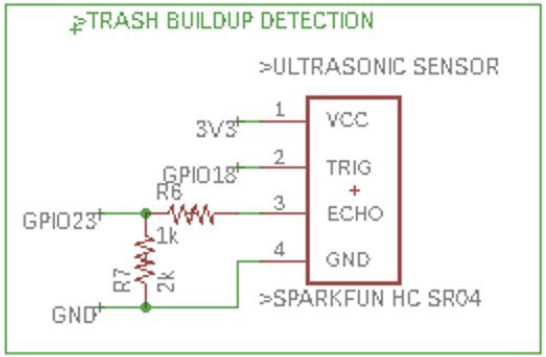


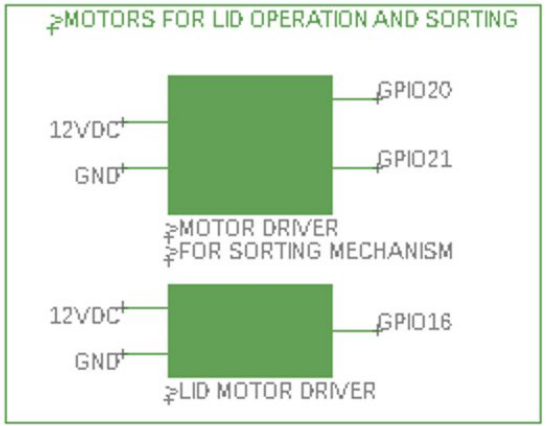


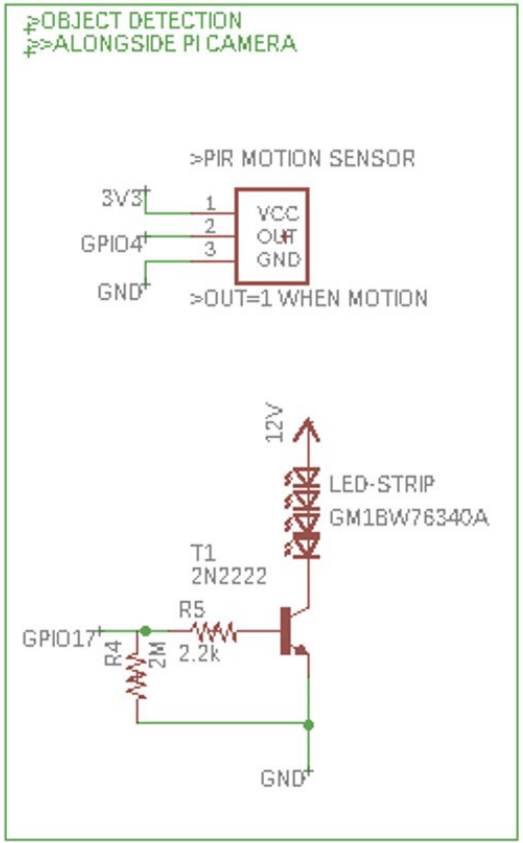
Hardware Schematic



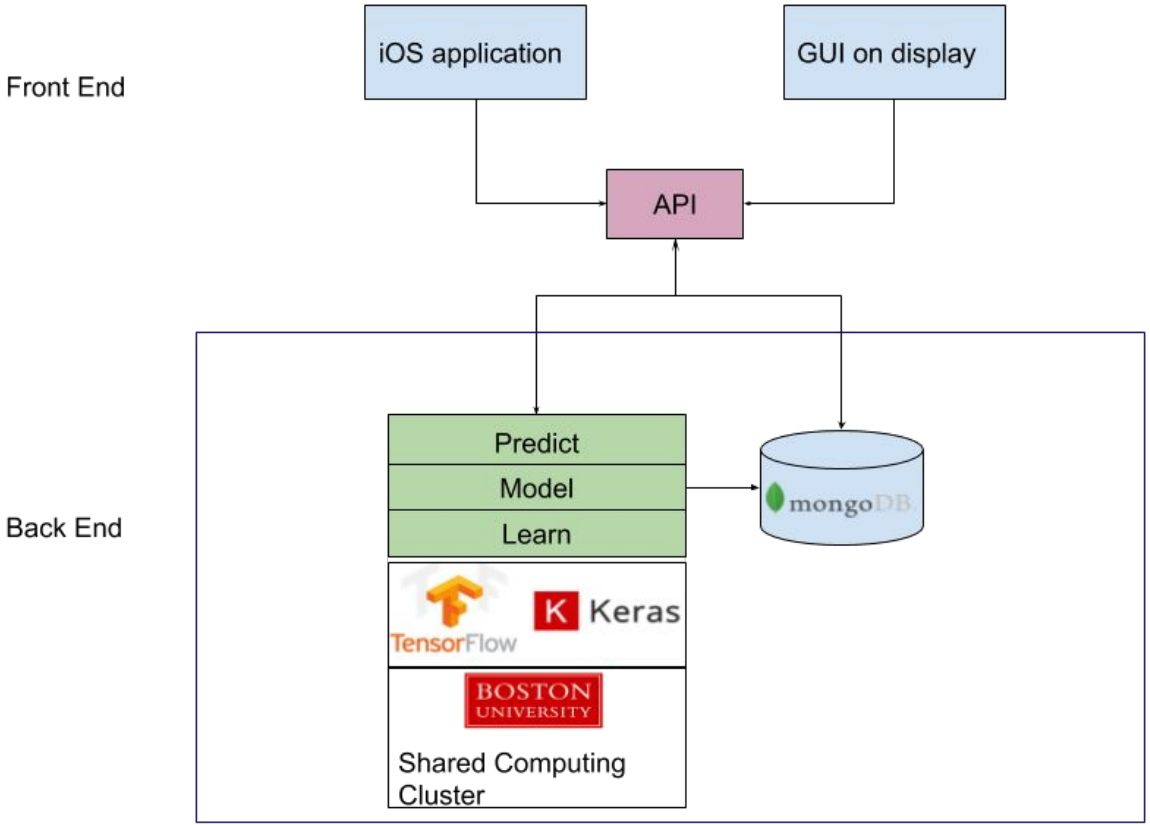








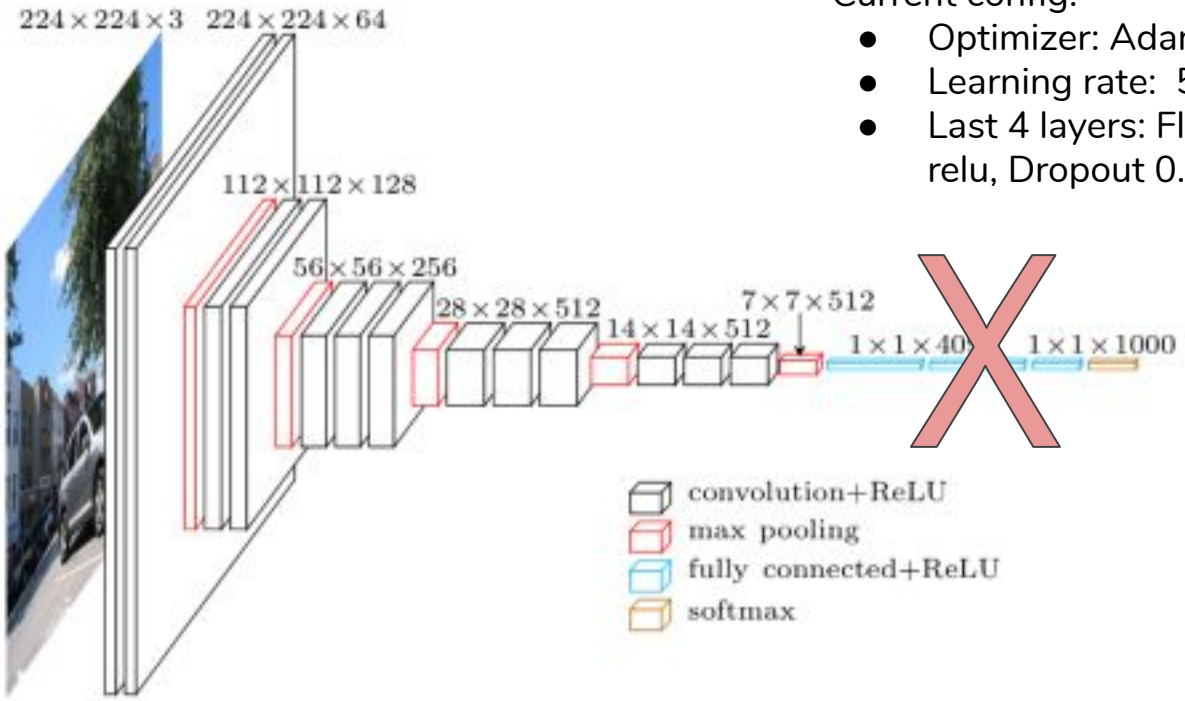
Software Diagram



Machine Learning Training & Results

- Powered by Keras and Tensorflow
- Fine-tuning on VGG16
- Deep Learning neural network with 16 layers
- 12 classes of objects
- Dataset collection with Google web scraper and Imagenet

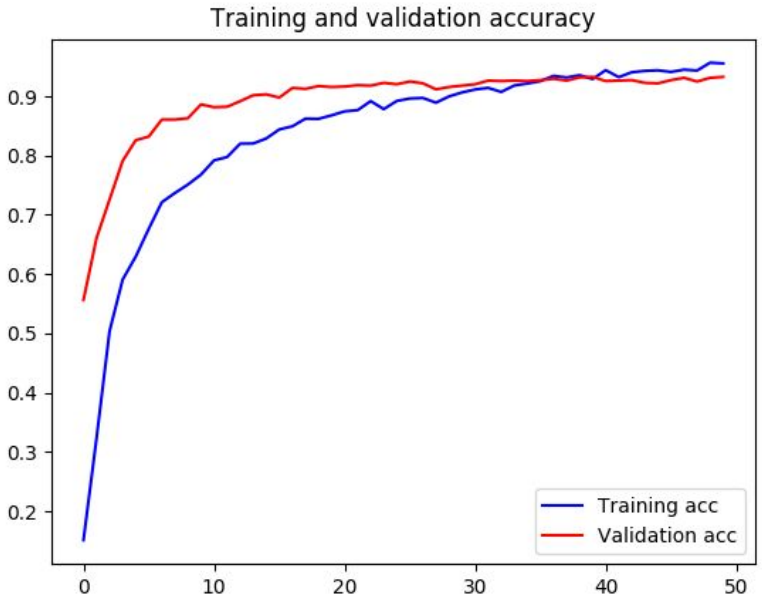
Machine Learning Training & Results

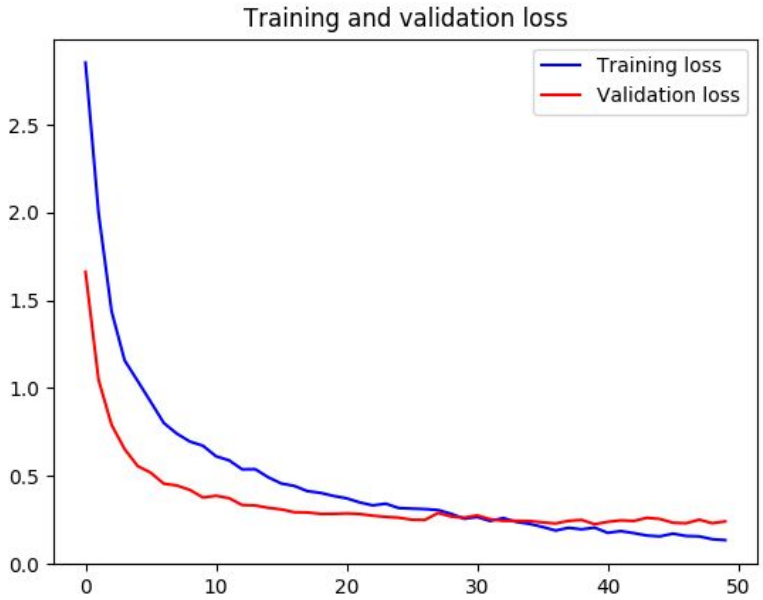


Current config:

- Optimizer: Adam
- Learning rate: 5e-5
- Last 4 layers: Flatten, FC 4096 relu, Dropout 0.8, Softmax 12

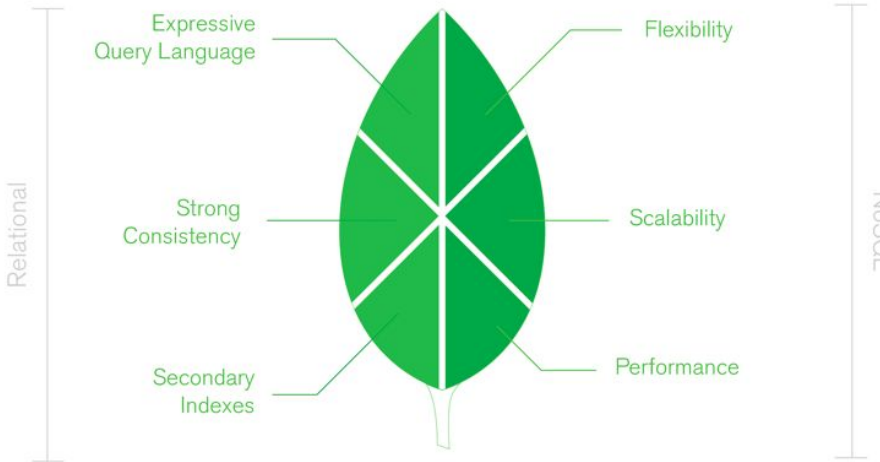
Machine Learning Training & Results





Database Architecture

- 2-Tier Architecture
- Flexible Schema
- 2 Document types: ***Temp*** and ***UserData***
- Temp Variables
  - Image Data (binary)
  - Result and Confidence Percentile
  - Flag
- UserData Variables
  - Weekly data
  - Opt: Location



```
"_id": {
    "$oid": "5c760183118d262d045c35e2"
},
"user_id": "temp",
"img": "",
"flag": "0",
"result": "",
"percentage": ""
```


- MLab Cloud Database
- Synchronization with IOS application
- <0.2s processing time


```
{
    "_id": {
        "$oid": "5c75fab9118d262c5f8496cc"
    },
    "user_id": "User1",
    "2_23_2019": "orange*apple*paper*plastic_bottle*paper",
    "2_24_2019": "paper*paper*orange*can*apple*glass_bottle"
}
```

GUI display

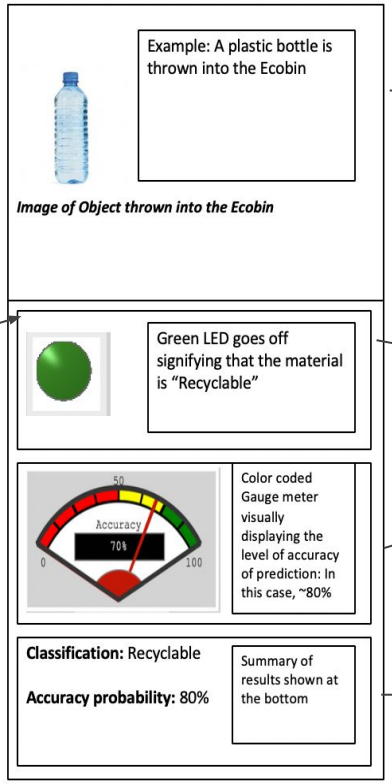


Image that is captured when placed into the ecobin

Extension tk_tools to implement the visual display of “gauge meter” and “LED”

Pack results into frame with summary

Prototype Testing & Results

1. Able to **differentiate between common cases** of Recyclables Trash
2. Success rate of **83%** based on 12 unique cases.
3. The loss of 17% came from **excessively white background** detection
4. **Most common anomaly** from external casing which resembles a “water bottle”

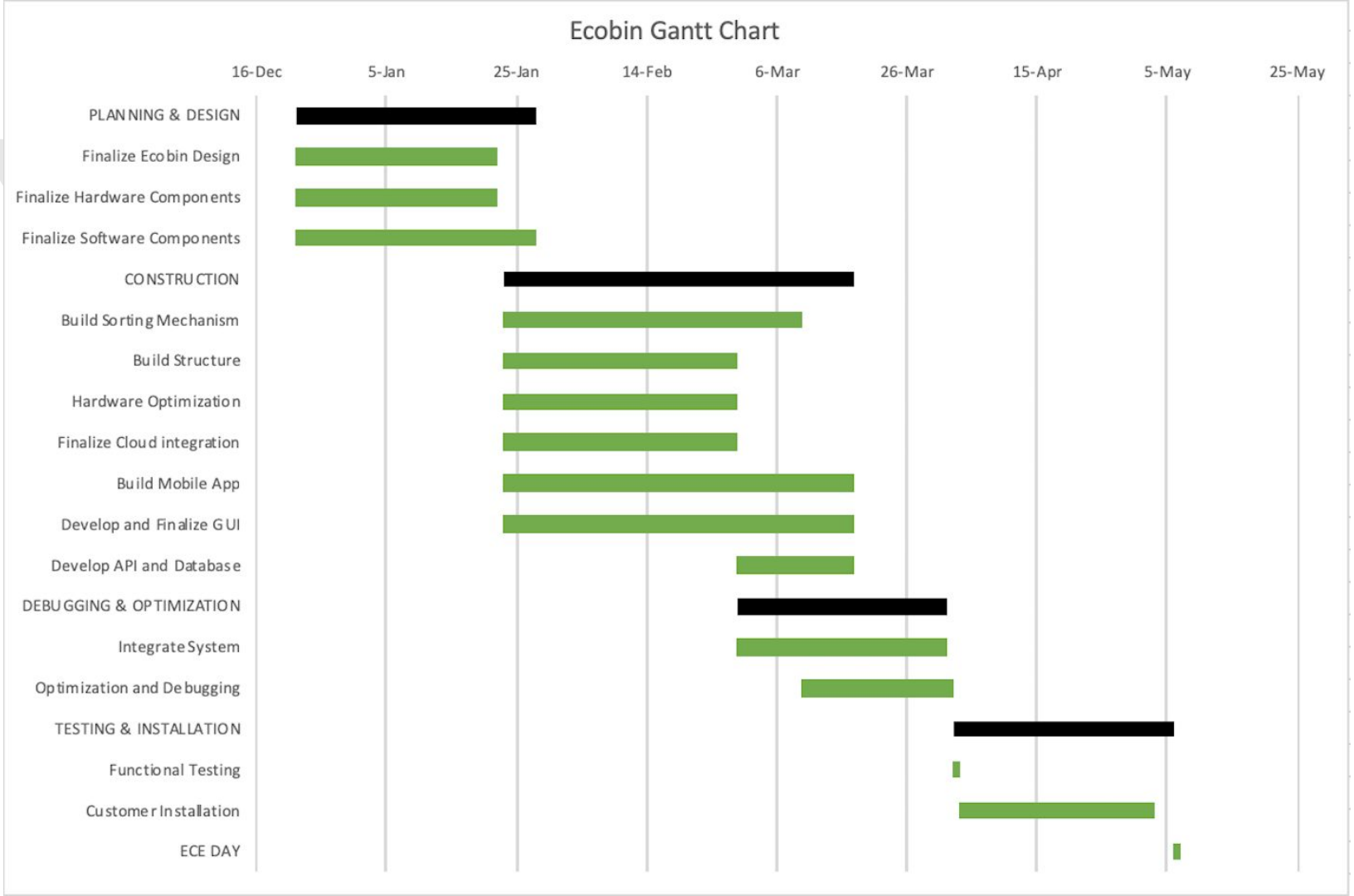
Future Goals & Expectations

Hardware:

- Use VNC instead of HDMI to display the GUI output
- Implement the LCD screen in the final product
- Configure power supply for all hardware (sensors, RasPi, motors, LED strip, etc)
- Reposition camera angle and position to capture better picture

Software:

- ML Model: Increase the number of classes to 15. Train more classes of objects for a more comprehensive classification system
- Build and apply a RESTful API (Flask)
- iOS application
- Optional: Remove background noise using HSV/OpenCV methods to ensure that it does not detect unnecessary features


Ecobin Gantt Chart
16-Dec
5-Jan
25-Jan
14-Feb
6-Mar
26-Mar
15-Apr
5-May
25-May
PLANNING & DESIGN
Finalize Ecobin Design
Finalize Hardware Components
Finalize Software Components
CONSTRUCTION
Build Sorting Mechanism
Build Structure
Hardware Optimization
Finalize Cloud integration
Build Mobile App
Develop and Finalize GUI
Develop API and Database
DEBUGGING & OPTIMIZATION
Integrate System
Optimization and Debugging
TESTING & INSTALLATION
Functional Testing
Customer Installation
ECE DAY

Thank You