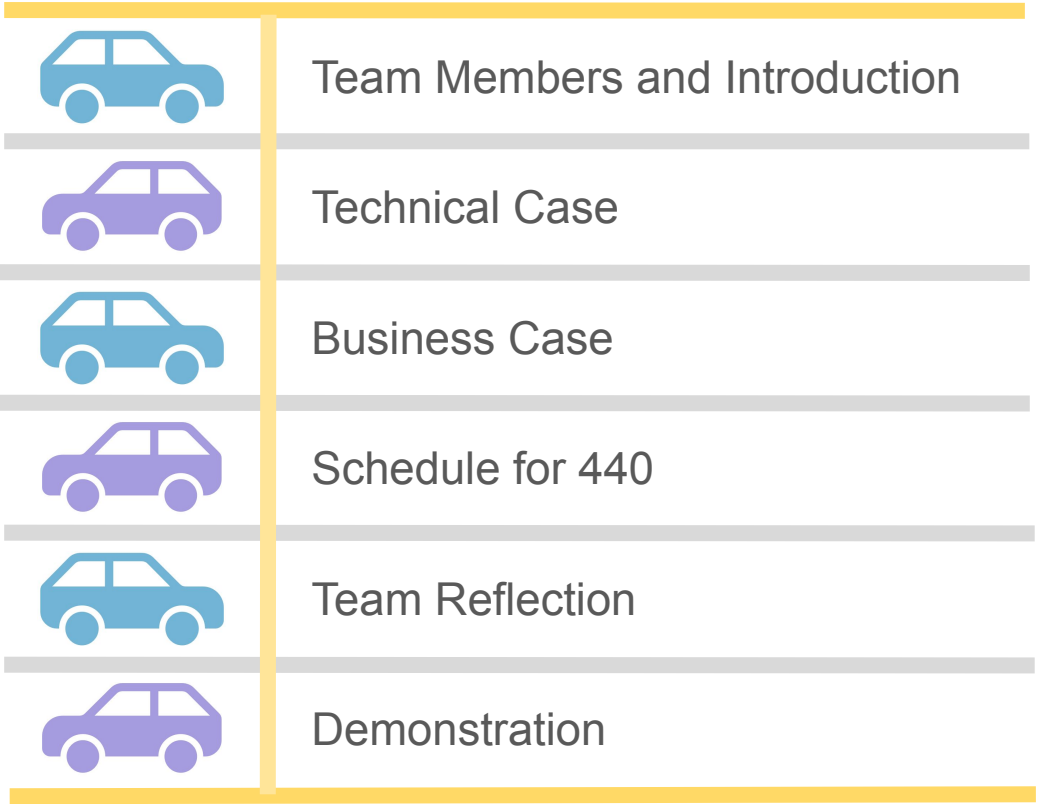
OpenSpot

PoC Presentation

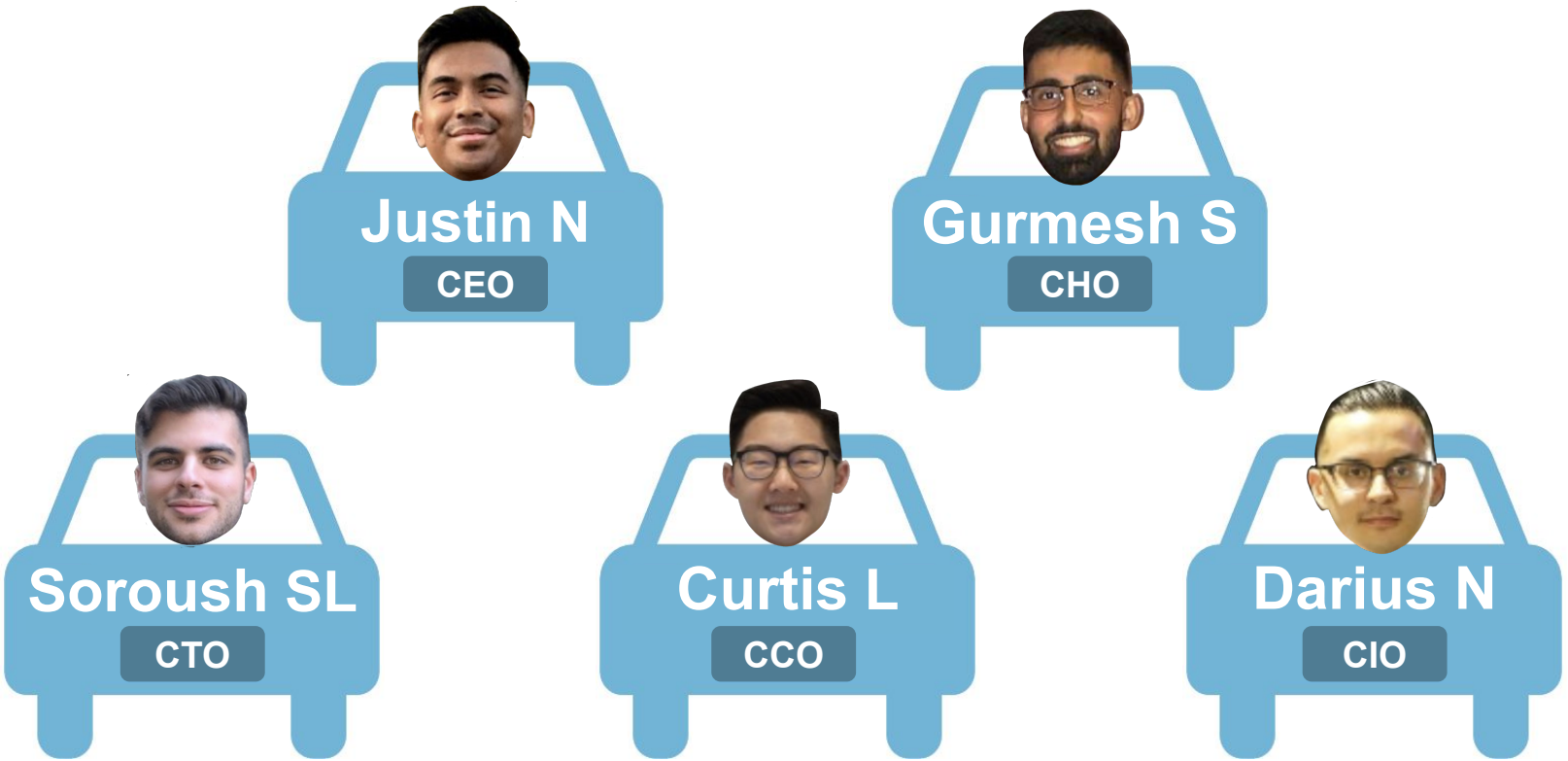
Presented by Company 7

OpenSpot

Presentation Outline



Team Members and Roles



OpenSpot

The Problem

Busy parking lots

Parking Disputes

Wasted Time

Our Solution

Camera locates open spots

Mobile-friendly website to browse parking areas with open spaces

LED indicator lights to show parking density in the area

High Level Overview

Hardware

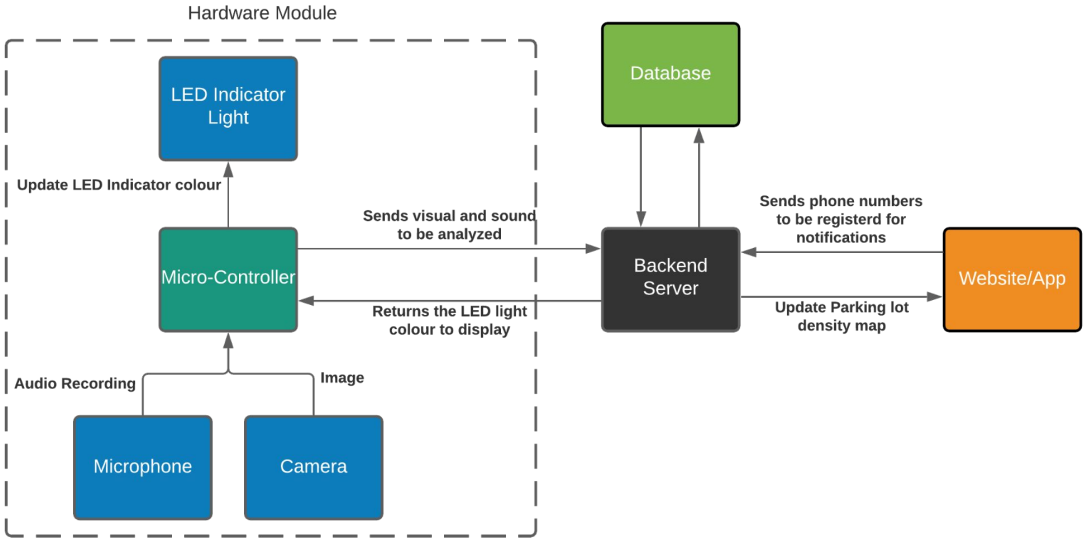
- Raspberry Pi 4
- Raspberry camera
- RGB LED
- Microphone

Backend Web Server

- Django Framework
- MongoDB Database
- Cloud Server

Computer Vision Model

- Mask R CNN Model with Pre-Trained Weights
- Trained on MS COCO Dataset



Main Functionalities

ENSC 405W

- Vehicle detection and parking spot vacancy prediction
- Subsystem communication
- Basic LED light indicator control

ENSC 440

- Website
- Car Alarm Detection
- Upgraded Hardware and Housing

OpenSpot

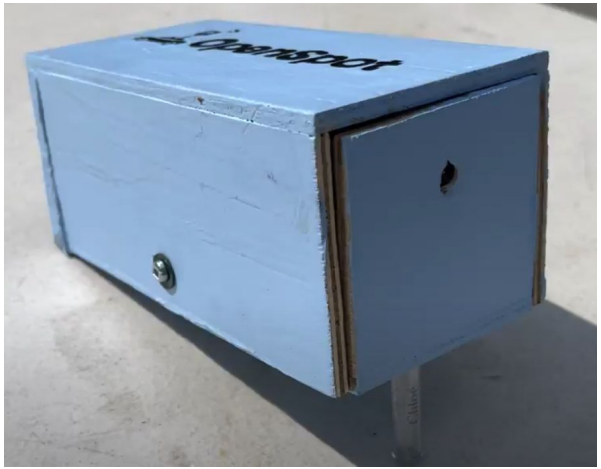
Materials and Sustainability

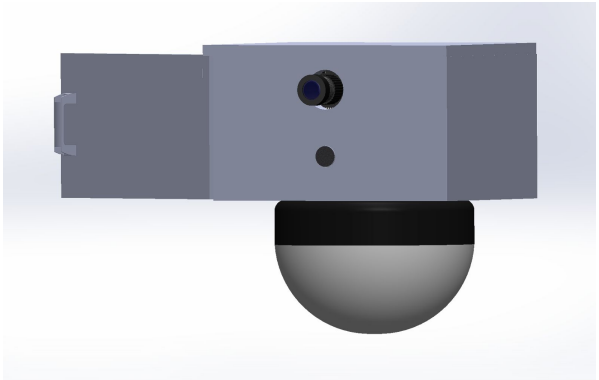
1. Housing is made of wood for PoC

2. All electronics used are reusable
   - Electronics can be recycled by ERA
   - Pi can recycled/reused using a free service called PICYCLE
   - Camera, microphone, and LED need to be broken down to recycle

Physical Design Changes

- Housing presented in design spec has changed
  - Modular design was created for the PoC
  - Shipping issues with our DMX light was a main factor for the enclosure change
  - Current housing accounts for smaller LED and a large breadboard



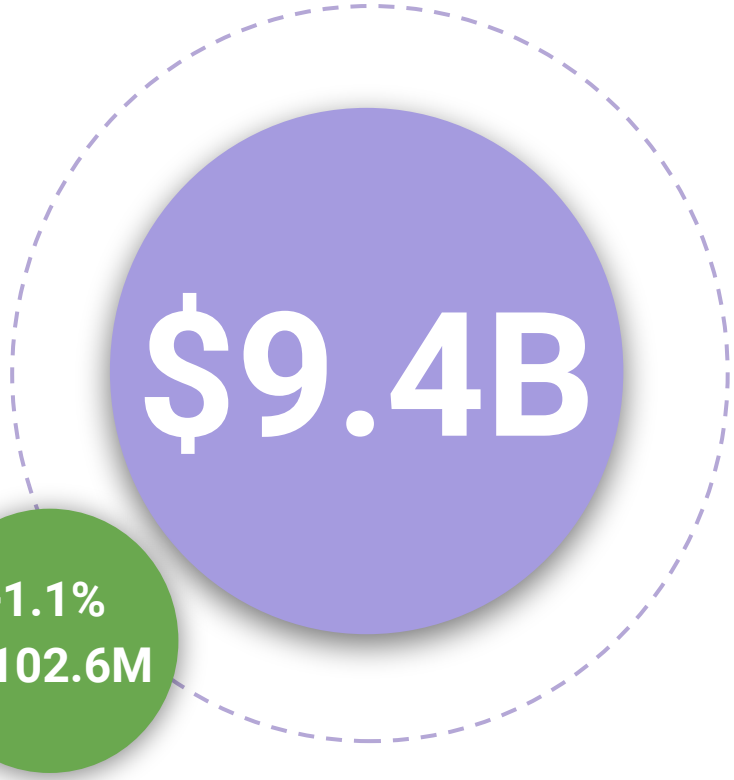


Software Design Changes

- Does not require a fully filled parking lot for a reference photo
  - We can use any reference photo and hand draw the bounding boxes
  - This change provides more accuracy in detecting open/empty spots

Market for Parking Lots

2021 Market Size



2021 Growth Rate

Note: Data is specific for Canada and United States [1][2]

OpenSpot

Market for Smart Parking Systems

Global Smart Parking Market

$4.4B

in 2020

Compound Annual Growth Rate

+21.5%

From 2021 - 2028

Parking Guidance System Revenue Share

over 75%

in 2020

[3]

Target Market

Outdoor uncovered parking with pre-existing light poles

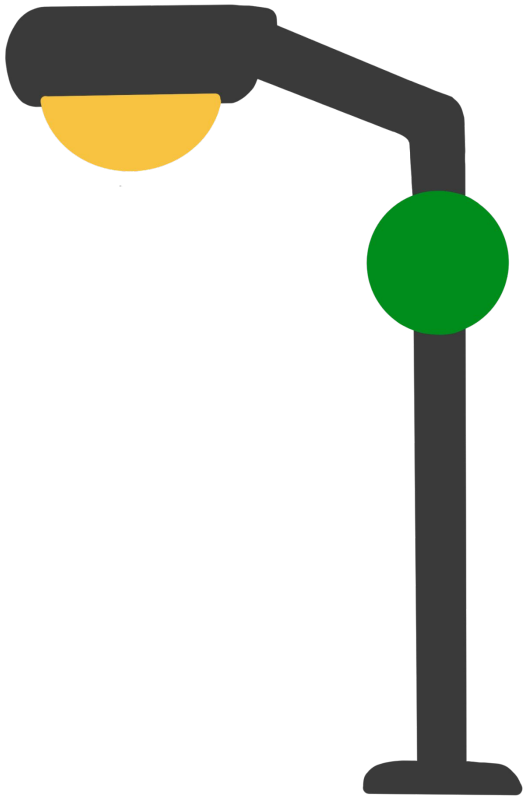
Main target market: universities and malls

Expand towards private parking lot owners and parkades

Customer Considerations

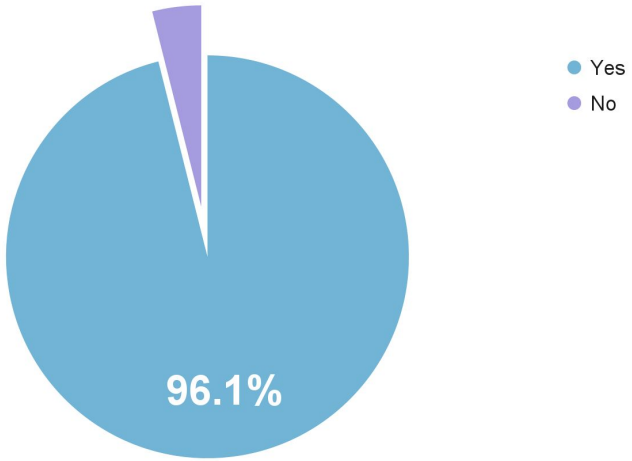
Build off of existing infrastructure

Per spot indication is unattractive and expensive

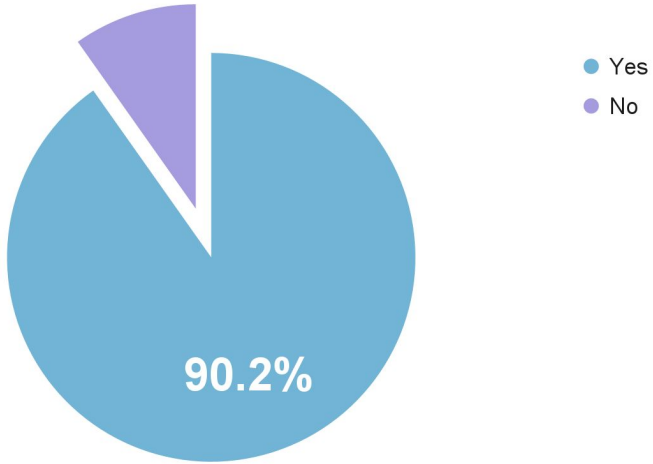


User Considerations

Would an indicator (physical lights) that displays empty parking spots in a parking lot be helpful when you are trying to find a parking spot?



Would receiving notifications (text messages) in advance that a parking lot is full/empty be helpful?



Competitors

https://www.smartparking.com/

https://stanley-robotics.com/

https://www.cleverciti.com/

Competitive Advantage

1. Not a per spot system, can cover multiple spots with a single module
2. Can be applied to any outdoor parking lot with pre-existing infrastructure
3. Provides Surveillance and security aspect
4. Also provides “guiding signage” with the module itself in the form of LED indicator
5. Have a website where motorists can plan their trips and monitor parking lots

OpenSpot

Budget Breakdown

PoC Prototype

Total Cost: **$227.28**

Parts:

1. Raspberry Pi 4
2. Raspberry Pi Camera
3. Housing Supplies

Engineering Prototype

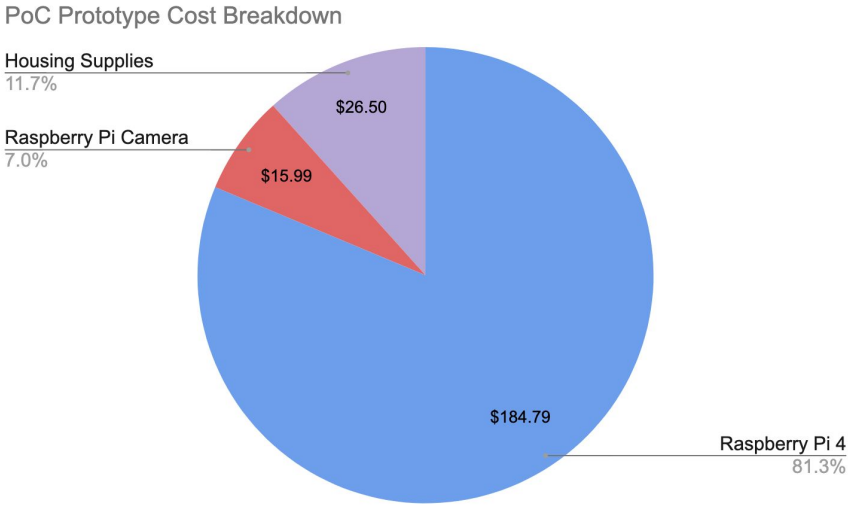
Estimated Total Cost: **$396.24**

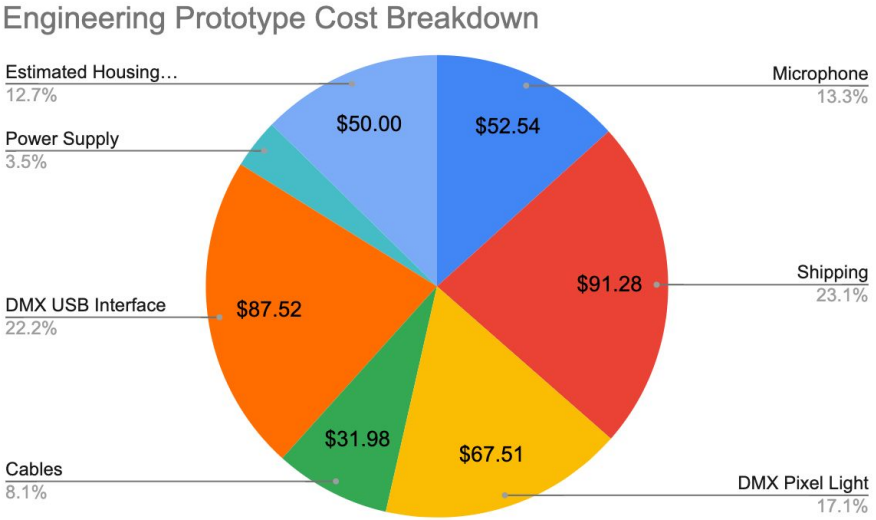
New Parts:

1. Microphone
2. LED Pixel DMX Dome Light
3. Power Supply
4. DMX USB Interface
5. Various Connector Cables
6. Housing Supplies

OpenSpot

Cost Comparison





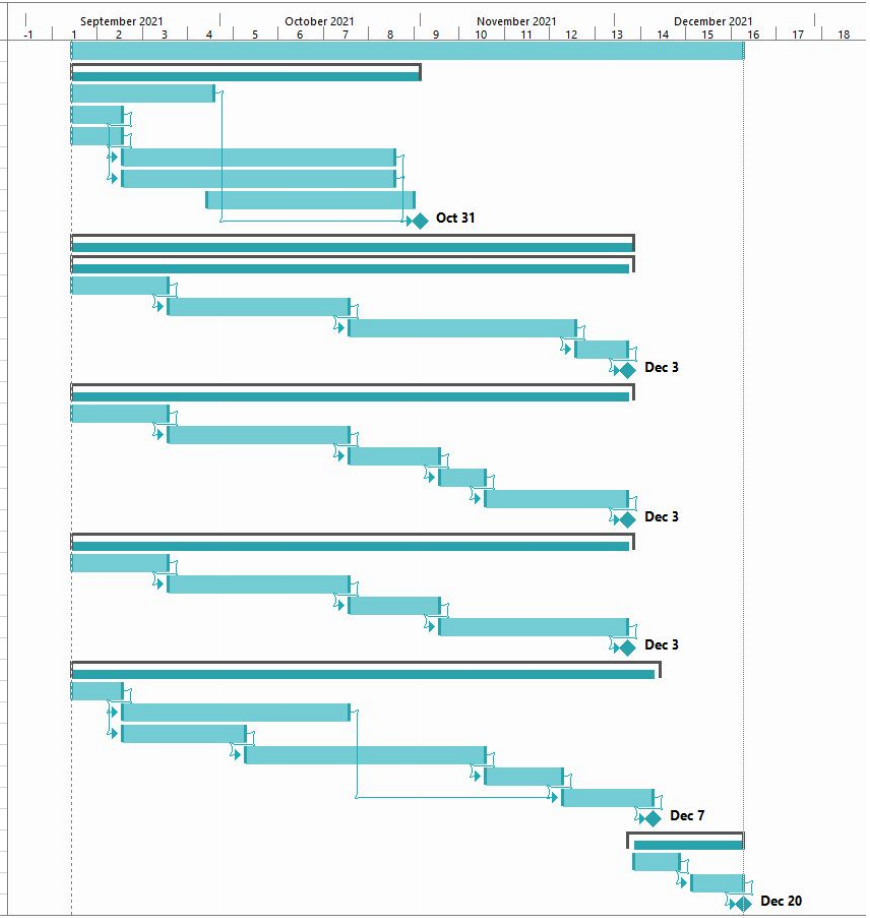
Total PoC Cost: $227.28

Estimated Eng. Prototype Cost: $396.24

Project Schedule

OpenSpot

ID	Task Name
1	**1 ENSC 440 Assignments and Tasks (TBD)**
2	**2 Hardware Module Development**
3	2.1 Model and Prototpye Different Housing Options
4	2.2 Connect New Lights
5	2.3 Set up microphone
6	2.4 Develop Script to Control DMX Lights
7	2.5 Develop Script to Control Microphone
8	2.6 Create Final Housing Prototype
9	2.7 Hardware Portion of Project Complete
10	**3 Backend Server Development**
11	**3.1 Computer Vision Application**
12	3.1.1 Gather Resources and Data to Train a Model for Difficult Weather
13	3.1.2 Set Up and Train Proprietary Model
14	3.1.3 Testing Proprietary Model
15	3.1.4 Final Development and Testing
16	3.1.5 CV Model Development and Training Complete
17	**3.2 Audio Detection Application**
18	3.2.1 Research and Gathering Data
19	3.2.2 Develop Audio Detection Script
20	3.2.3 Test Audio Detection App
21	3.2.4 Final Development and Testing
22	3.2.5 Develop Script for Website Notifications
23	3.2.6 Audio Detection App Complete
24	**3.3 Text Notifications**
25	3.3.1 Research 3rd Party Choices
26	3.3.2 Develop Server Script to Communicate With API
27	3.3.3 Testing
28	3.3.4 Final Development and Testing
29	3.3.5 Text Notification System Complete
30	**4 Website Development**
31	4.1 Main Home Page
32	4.2 Map API
33	4.3 Login and Registration
34	4.4 Text Notification Sign Up
35	4.5 About Us/Contact Page
36	4.6 Testing
37	4.7 Website Hosted and Complete
38	**5 Integration and Testing**
39	5.1 Executing Test Plan
40	5.2 Bug Fixes and Final Tests
41	5.3 Final Eng. Prototype Complete



Lessons Learned

Hardware

- Communicating with sales representatives for products in advance
- Designing the housing unit with collective input from the OpenSpot team
- Establishing communication with the Raspberry Pi 4 as early as possible
- Having standby options available for the power source used by the OpenSpot module

Software

- Testing communication with the Raspberry Pi 4 to modify the LED light earlier
- Learning that the origin used in photos started at the top left corner
- Unrealistic standard of execution time for computer vision application

Changes for ENSC 440

Ensure that deadlines to complete individual tasks are upheld

Have more group members work on the backend server aspect of our system

Meeting more frequently during the development of our website

Demo Video

OpenSpot

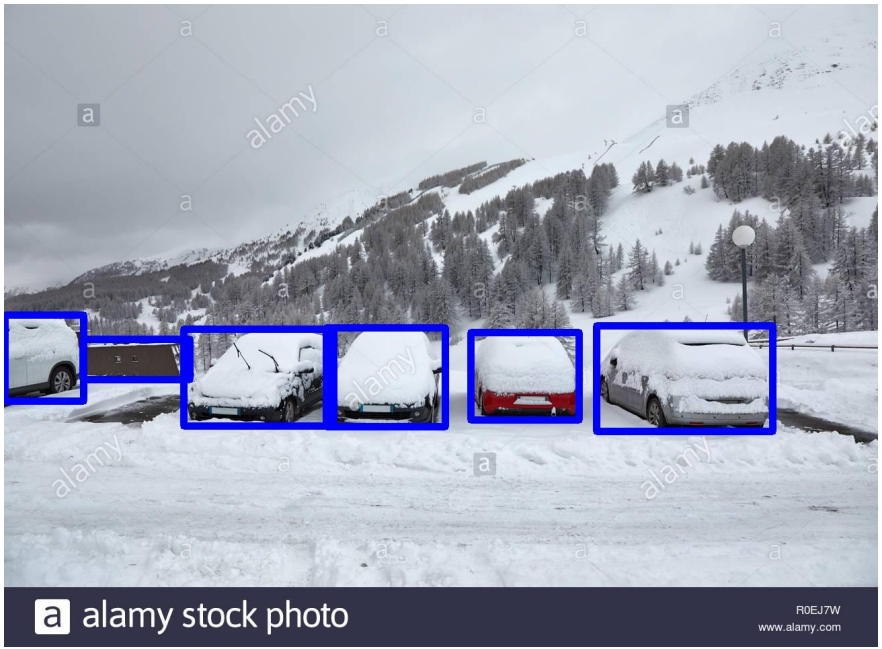
ACCEPTANCE TEST PLAN

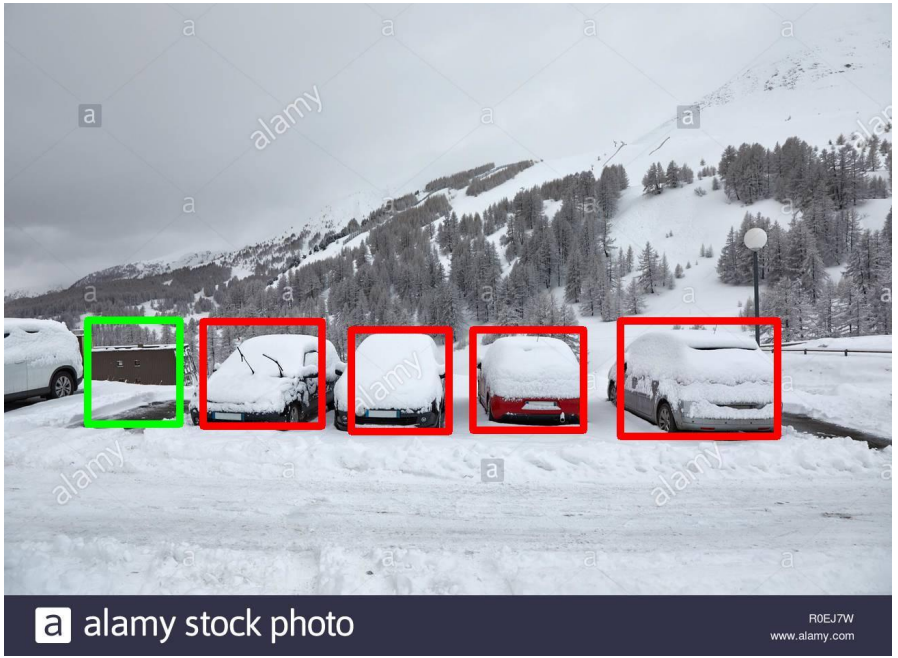
Stage: *Alpha Prototype*

Purpose	Test Description	Acceptance Criteria
Validate connection between microcontroller and peripherals	Microcontroller takes a picture using the camera. Microcontroller changes the colour of the LED.	Image is viewable as a local file on the microcontroller. Microcontroller must be able to change the colors of the connected lights.
Validate local server connection with microcontroller	Apply tests of sending various data to and from the microcontroller to the backend server. Server can respond with what the LED colour should be.	The local backend server receives the file. The local server can tell the microcontroller to change the LED colour.
Validate system can correctly detect empty and taken parking spots	Input various images of parking lots into the system. Images should range in size of parking lots and amount of cars.	The system should be able to correctly identify where the parking spaces are, which objects are cars, and which parking spots are empty or taken.
Complete system test	Set up a module in a parking lot. Have the module take a picture and run it through the system once.	Module takes a picture and successfully sends the image to a local server. The local server is able to compute how many cars are in the image. Local server can tell the microcontroller what colour to set the LED.

Key Feedback From Progress Review Meetings

Weather conditions - Snow





OpenSpot

Question Period

References

[1] “Industry Market Research, Reports, and Statistics,” IBISWorld. [Online]. Available: https://www.ibisworld.com/canada/market-size/parking-lots-garages/. [Accessed: 02-Jun-2021].

[2] “Industry Market Research, Reports, and Statistics,” IBISWorld. [Online]. Available: https://www.ibisworld.com/industry-statistics/market-size/parking-lots-garages-united-states/. [Accessed: 02-Jun-2021].

[3] Global Smart Parking Systems Market Size Report, 2021-2028. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/smart-parking-system-market. [Accessed: 02-Jun-2021].