Intersection Traffic Automation for Vehicles

William Valentine, Avery Belanger, Jie Zhao

Presentation Outline

- Introduction
- Assembling the car
  - Motors
  - Server
- Methods
  - Design of the Server and Car
  - The Re-creation of an Intersection Scene
  - YOLOv8
  - Car Identification
  - Car and Pedestrian Queuing
- Results
  - Demo
- Future work

What is Computer Vision

Source: Narc — A Fault Detecting Solution, Medium

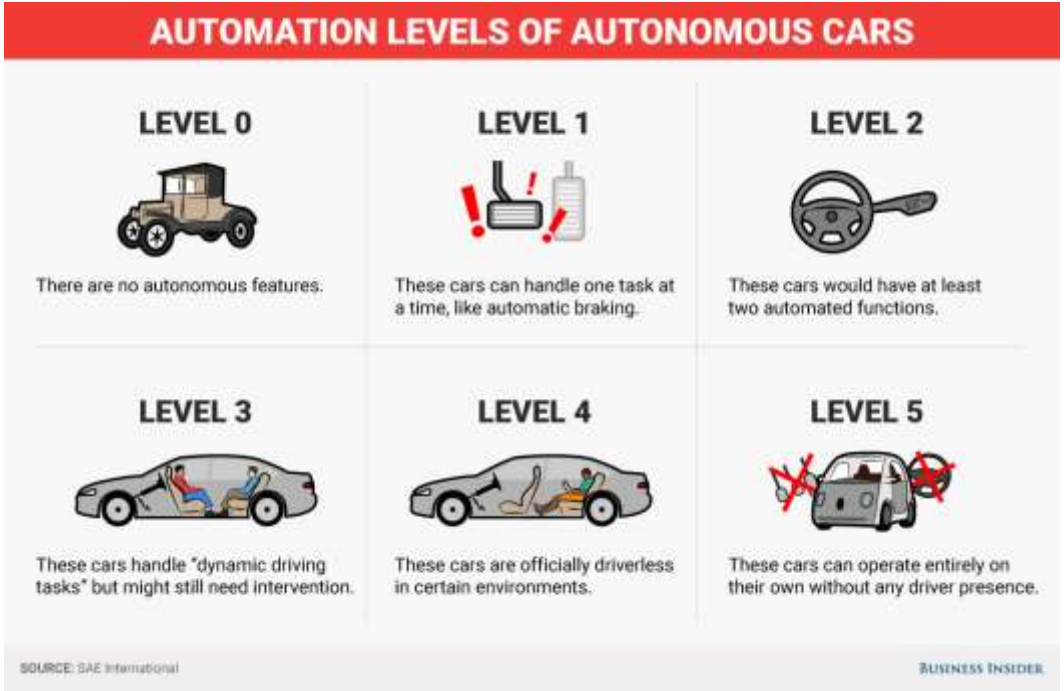
What is Computer Vision

- Two main parts of our studies
  - Image recognition
  - Object detection

- Image recognition
  - You look at an image of fruit and recognize an orange in the image

- Object detection
  - You look at an image of fruit and determine where an orange is

What are Autonomous Vehicles



Source: Business Insider

What are Autonomous Vehicles

The components of autonomous vehicles can be broken down into two categories.

- Hardware
  - Traditional automotive parts (engine, brakes, and wheels)
  - Sensors (LiDAR, Radar, and Cameras)
  - On-board computers (for analyzing complex information in real-time)

- Software
  - Network communication
  - Computer vision
  - Path planning

Research Motivation (Why this matters)

- Self-driving cars are expensive ($70,000-$150,000, Reuters)
- Intersection traffic is often an issue because people do not want to wait at intersections
- Intersections often cause the highest number of accidents on roadways (>50%, Federal Highway Administration)

Our Research Goal

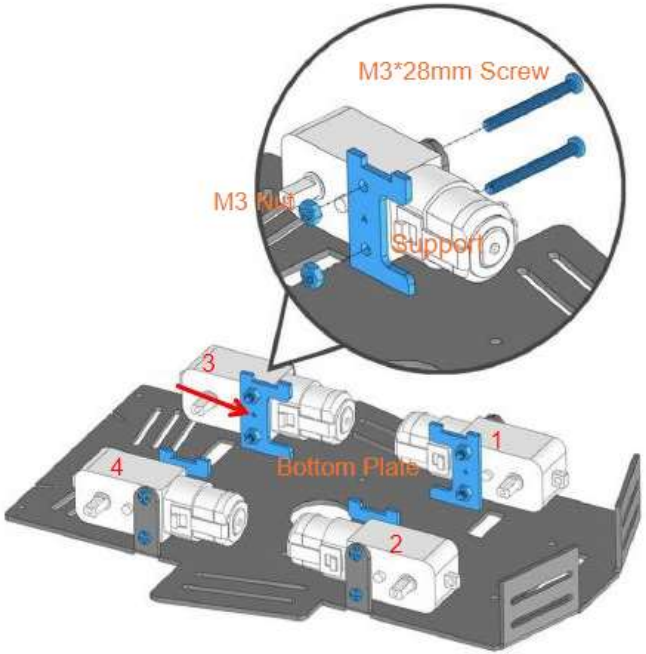
To mitigate the issues caused by intersections, we purpose creating a centralized control system of remote-controlled cars, to see if a real-world implementation of sensor-less vehicles for intersection automation is feasible.

Assembly of the Car

- Raspberry Pi Pico (running micropython)
- PICO RDP (Robotic Development Platform)
- ESP8266 ESP01s (WiFi module)
- Battery
- Four individual motors
- Radar sensor
- Servo (for radar sensor)
- Speed sensor
- Grayscale sensor

Source: Raspberry Pi Pico 4wd car, SunFounder

Motors



Source: Sunfounder-Picar-4wd, Manualslib

Server Specifications

- CPU: Intel i7-10700F @ 2.9GHz
- Memory: 16GB
- SSD: 512GB
- OS: Windows 10
- GPU: GeForce GTX 1660 Ti

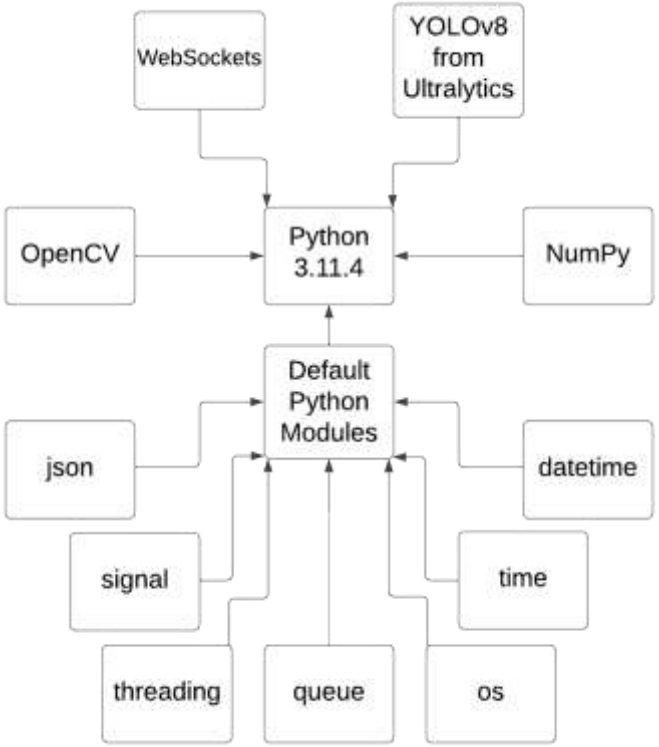
Source: Dell G5 Gaming Desktop, Amazon.com

Camera Specifications

- Name: Logitech C922x Pro Stream Webcam
- Maximum Resolution: 1920×1080
- Maximum Framerate: 30 fps (@ 1080p)

Source: Logitech C922x, Amazon.com

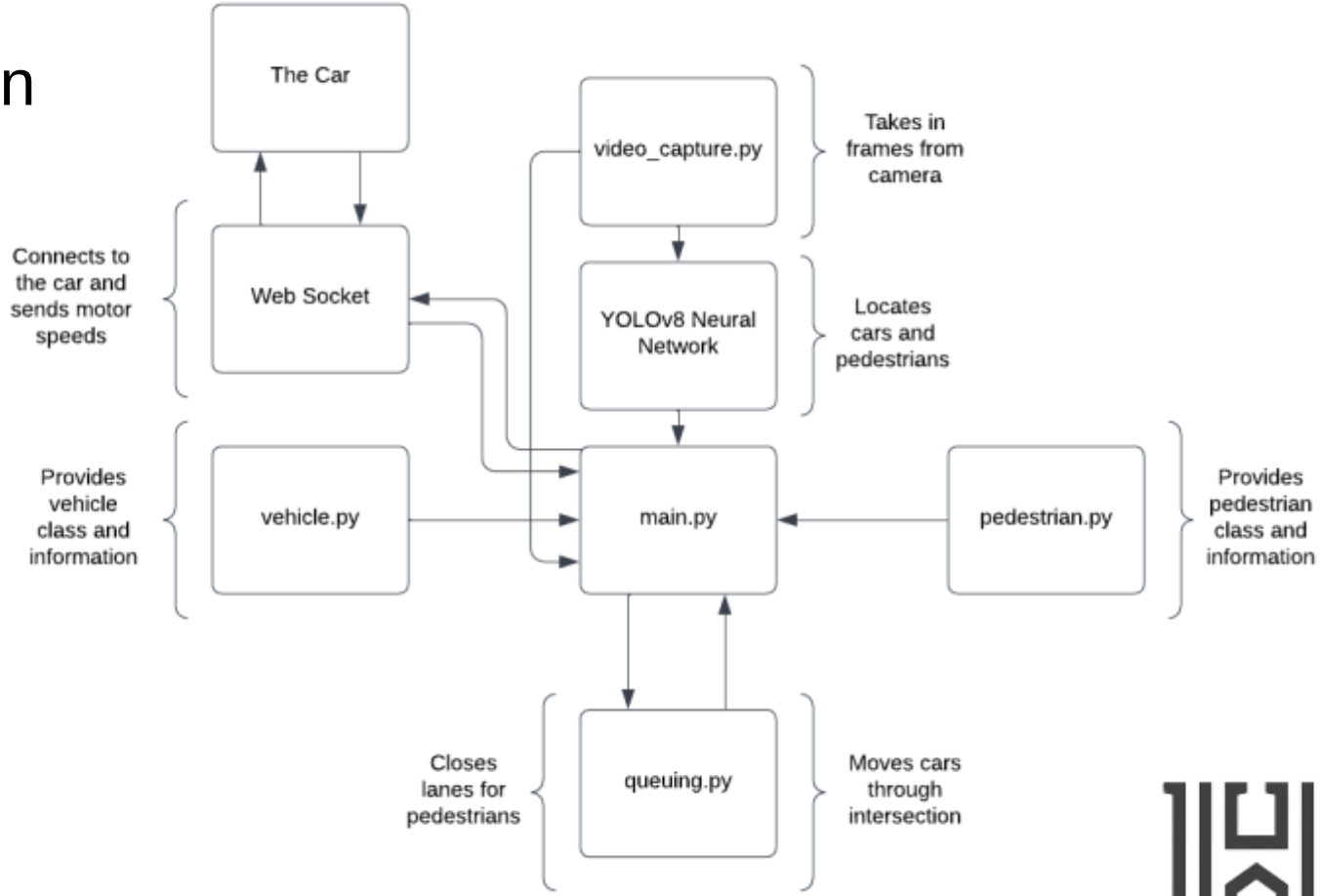
Server Programming Language and Libraries



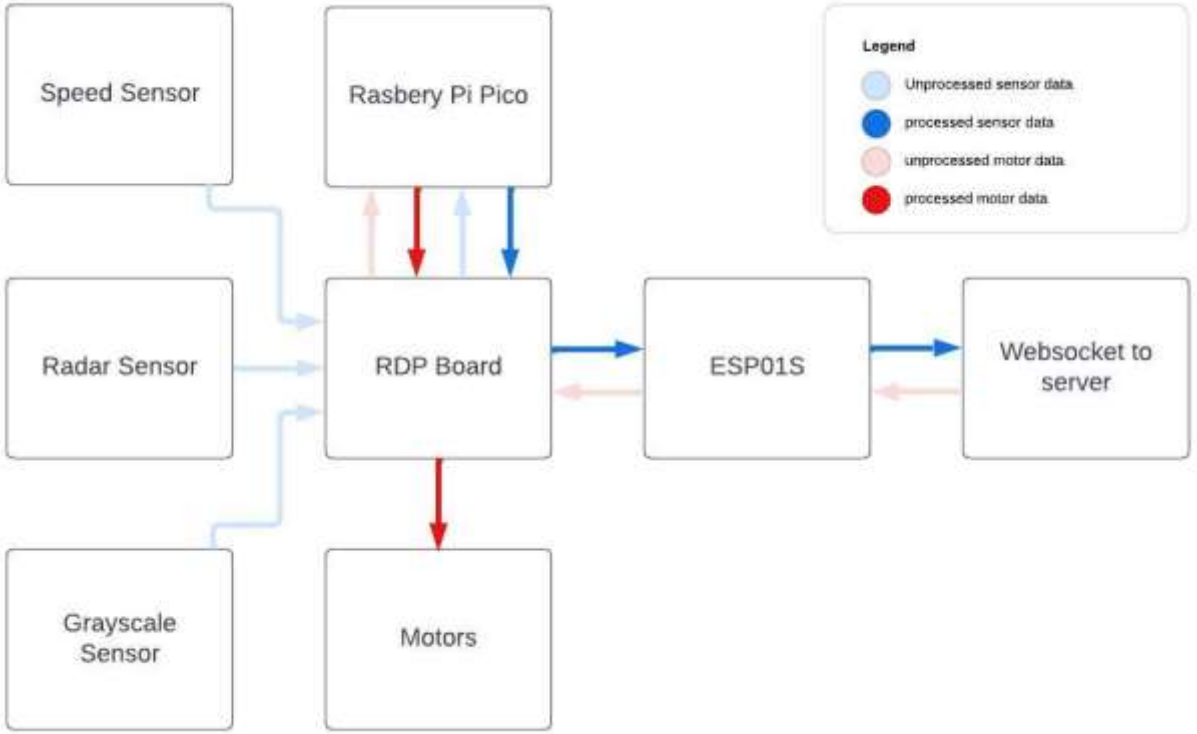
The Design of the Server and the Car

- The Server sends commands and operates the Car(s)
- The Car(s) receive commands and drive automatically
- Using websockets, the Server and the Car(s) are able to communicate with each other

Server Design

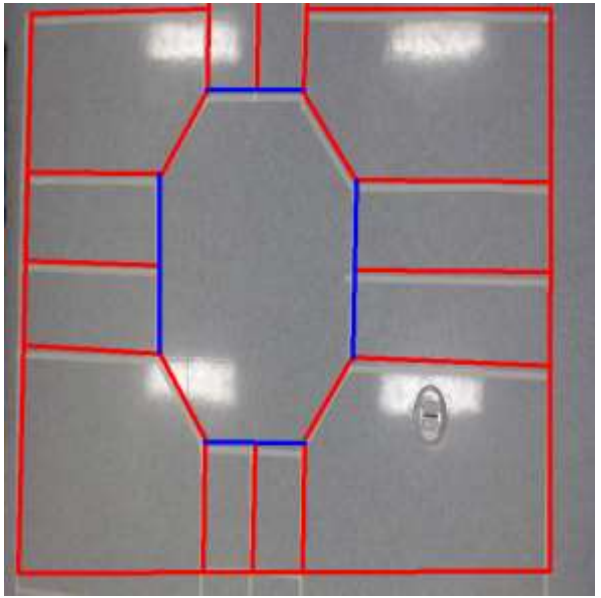


Car Design


Speed Sensor
Rasbery Pi Pico
Legend
Unprocessed sensor data
processed sensor data
unprocessed motor data
processed motor data
Radar Sensor
RDP Board
ESP01S
Websocket to server
Grayscale Sensor
Motors

Re-creation of an Intersection Scene

- Red lines represent the lanes and border
- Blue lines represent the stop lines
- These lines drawn over the image assist with lane detection and the queuing process



What is YOLOv8

- YOLO (You Only Look Once) is a popular Computer Vision Neural Network for detecting images in real-time frames

- We have employed a data labelling tool, Roboflow, to annotate images and then train a YOLOv8 model in a form of “transfer learning”

Source: Ultralytics YOLOv8, linkedin.com

Proposed YOLOv8 Model

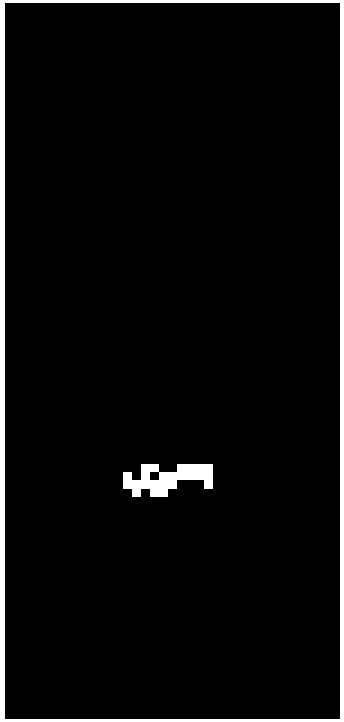
- Trained Epochs: 53
- Total Images: 29
  - Training images: 18 (62%)
  - Testing images: 5 (17%)
  - Validation: 6 (21%)
- mAP50: 99.5%
- mAP50-95 Total: 73.3%
  - Car: 85.3%
  - Pedestrian: 61.3%

Source: Roboflow

Car Identification

- To identify specific cars, the server uses color-based identification
- The server identifies each car by matching it to a unique color on a piece of paper, detecting the presence of this specified color based

Image of car based on bounding box location

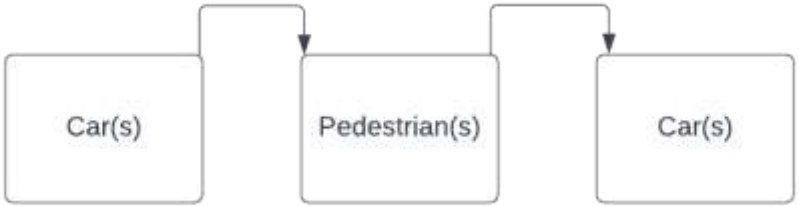


White pixels mean the color has been detected at these locations

Queuing System

- The queuing system has been developed for keeping track of the order of traffic
- For example, if a pedestrian wants to cross the road, the traffic timing will be allocated
- The queuing system will shut down the required lane for the pedestrian, keeping all other lanes open
- This will bring more efficiency to crowded intersections because cars will spend less time waiting

Queuing System Pseudocode

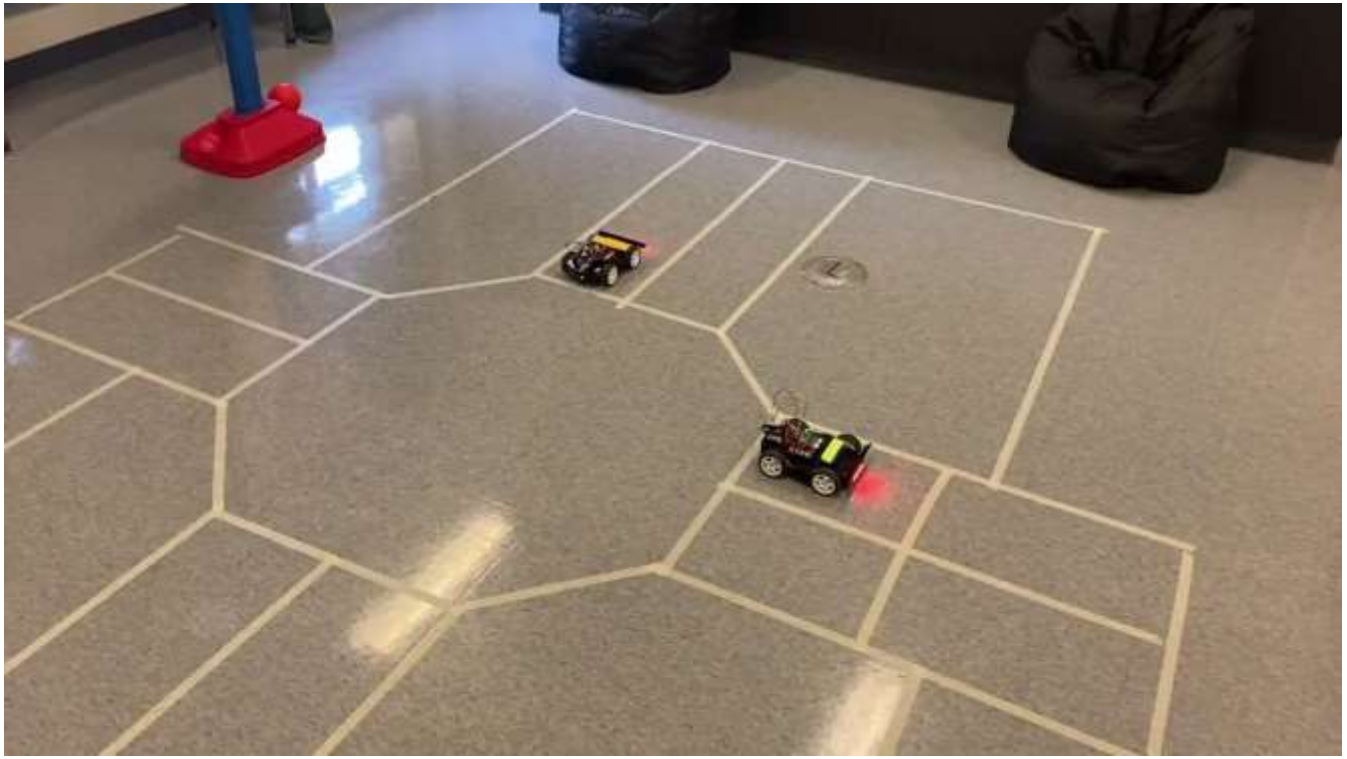



```
When an object is noticed by the camera:
     add to the queue;

While first item in queue is a Pedestrian:
         Close down lane that the Pedestrian needs for 15 seconds;
         Remove pedestrian from the queue;

Iterate through items in queue:
         If item is a car:
                  If the lane is open:
                           Car moves;
                  Else:
                           Car doesn't move;
```

Demo: Multiple Cars Turning



Demo: Car Waiting For Pedestrians

Research Outcome

Based on the testing results, we believe that the automatic intersection we designed would be feasible at a larger scale in the future.

Future Research

- Implementation of real-world vehicles in a simulation
- Developing a more robust driving system
- Responding to obstacles
- Training a larger and more robust object detection model
- Creating a robust real-world car identification system
- Running a model of both autonomous and manual vehicles at an intersection
- Using LiDAR along with a camera to avoid identification issues caused by lighting or weather