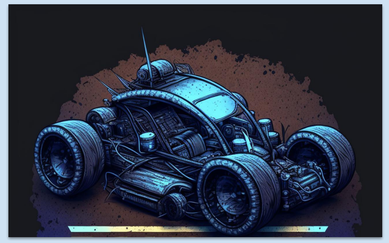
Modular Package for Autonomous Driving (mPAD)

Rohan Anand, Martin Bleakley, Allison Colón-Heyliger, Ian Khung, Ishan Rathi
Advisors: Kaveh Pahlavan (ECE/CS) & Pradeep Radhakrishnan (ME/RBE)

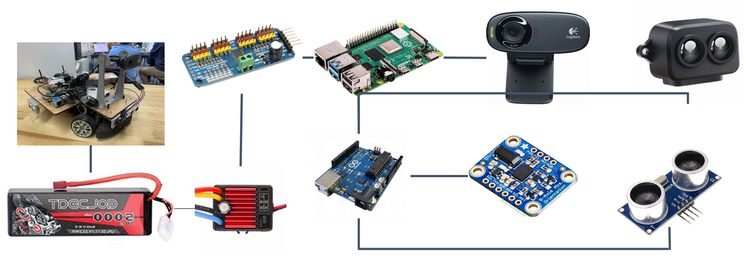


Project Overview

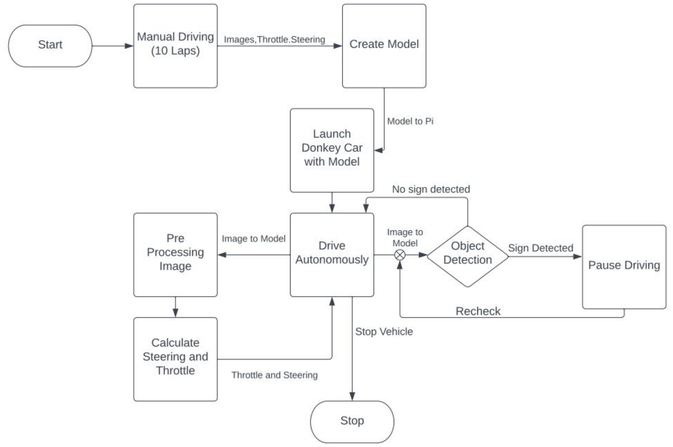
- The objective of the mPAD project is to develop a modular, open-source autonomous driving solution for scaled cars.
- The solution utilizes a package consisting of a Raspberry Pi, an Arduino Uno, two USB cameras, and multiple sensors.
- The project also integrates an intuitive web-based dashboard.
- The mPAD system provides a comprehensive, reliable, and easy-to-assemble autonomous driving solution for scaled cars.
- Users install this software onto their device(Raspberry Pi) and manually train about 10 laps around a track
  - This data is passed into a program that creates a model
  - Model is uploaded to the Raspberry Pi and launched using Donkey Car command line
  - RC car is then ready to drive autonomously using the new web-based dashboard

Hardware Design

- Several different styles and sizes of car were utilized.
  - Both pre-built and student designed remote control cars were implemented.
  - All cars use front-wheel drive with servo steering
- The sensor housings and camera mount were 3D printed out of PLA.



Driving Logic



Before

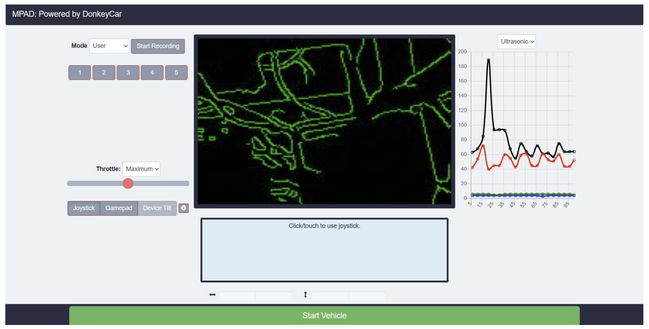
After

Results

- System capable of running 5+ laps on multiple different RC cars
- Detects a stop sign and successfully stops
- Preprocesses an image to normalize effects such as light and line color
- Updated dashboard with a new modern looks and incorporated data collection graphs

Dashboard

- Reorganized the layout of Donkey Car Dashboard
  - Camera is in the center of dashboard
  - Throttle and steering mode is on the left column
  - Graphs of sensor data are on the right column
    - 4 Ultrasonics, 1 IMU, 1 TF Luna
- Changed the color palette for the dashboard
- Incorporated three different driving modes
  - Manual, Autonomous, Auto Steer



Object Detection

- Optimized multiple versions for maximum frames per second.
- Utilized multiple PI's with a communication protocol between them(GPIO): ~20 fps
- Extended Donkey Car's capabilities to incorporate the communication protocol with a second Raspberry Pi
- Detected and stopped at a stop sign in real time with an accuracy of 84% using a CNN model
  - Input - A scaled 128 x 128 image from the webcam stream
  - Output - A boolean indicating a stop sign is seen