Stop Light AI

Andrew Dillon, Kentaro Matsuo, Lucas Gaudet, Samuel Bernsen

Abstract

- **Our Challenge: Harness AI for Safety Purposes**
- **Bettering Society with Safer Roads:**
  - Reducing the number of vehicles that run red lights (reducing red light accidents)
  - Safely and accurately determine if it is safe to "make a yellow light"
  - Overall reducing the amount of human error
- **Progress:**
  - Built a printed miniature vehicle using 3D printed parts, a raspberry pie, motors and sensors
  - Used computer vision to detect colors
  - Accessed pre-trained data set

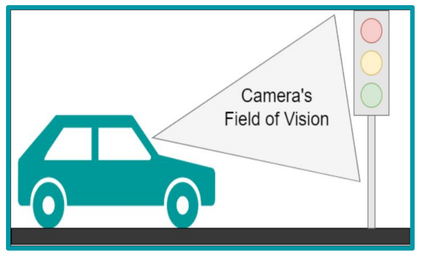


Figure 1: Visual representation of how our model would work

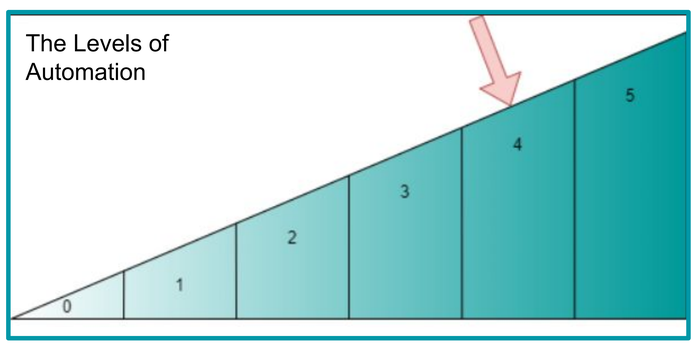


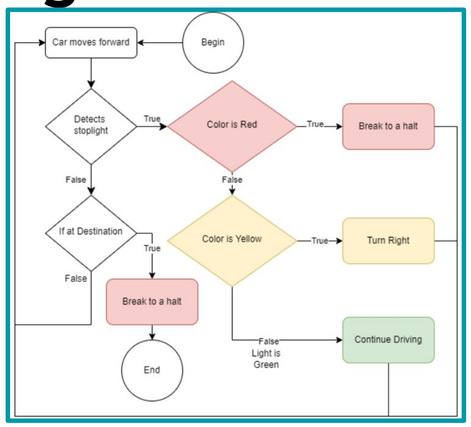
Figure 2: The 5 levels of Automation a based with an indication of where our model would fall on this spectrum [1].

Introduction

- **Background**
  - Autonomous cars are capable of sensing its environment and operating without human involvement
  - Reducing human error -> reducing accidents
  - Using sensors, actuators, complex algorithm, machine learning systems and powerful processors
  - Increasing safety, productivity, cost, efficiency, and accessibility
- **Our solution**
  - Stop light Recognition demonstrated on a small scale model of a car
  - Able to recognize stop lights in its field of view, identify which color light is currently activated and start or stop accordingly

Project Stages

- **Planning Stage**
  - UML Diagrams and Flowcharts
  - Designing and sourcing parts to 3D print
  - Research machine learning tools



Kornia
Open Source and Computer Vision
Powered by PyTorch

Figure 3, left: Some of the software we used for the project.
Figure 4 right: UML Flowchart detailing the steps our car needs to take when encountering a stop light

- **Build Stage**
  - 3D printed parts to attach some the hardware elements to better serve our needs
  - Built and programed stoplight to run tests with
  - Printing and resizing 3D parts
  - Programming, wiring, and soldering the stoplight
  - Attaching motor driver to raspberry pi
    - Soldering motors to 4 output pins
    - Driving motors with python library
  - Integrating motor driver program with machine learning algorithm

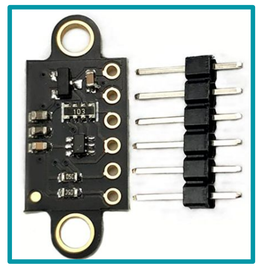


Figure 5 above: Photo of the Time of Flight sensor

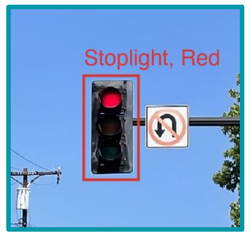


Figure 6: Zoomed in image showing how the stop light is labeled.

Citations

[1] Captcha. Osapublishing.org. [accessed 2022 Apr 13].

[2]Jeff Hecht. (2018). LiDAR for Self Driving Cars. Optics and Photonics News.

[3]Media U, Events. 2019b Jan 23. How anomaly detection is helping OEMs make autonomous vehicles safer. Automotive Testing Technology International. [accessed 2022 May 3].

[4]Baidac I. 2020 Nov 10. Sensor Fusion: a prerequisite for autonomous driving. The Autonomous. [accessed 2022 May 4].

[5]Amazon.com. [accessed 2022 May 9].

Set-Backs

- Car was damaged, snapping much of the original chassis and frame
  - 3D printed our own parts to replace what was broken and modify the original model to better fit our needs
- Ordered new ToF sensor
  - Not needed in final implementation
- Broken RGB camera
  - Ordered new camera to replace it
- Package manager needed for PyTorch
  - Difficulties downloading it to school computers due to not having the correct privileges. We were able to download it to our own laptops to begin work

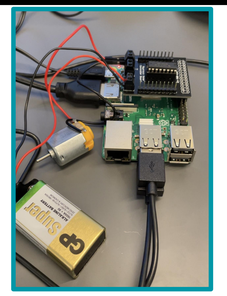


Figure 7 above: Raspberry Pi and DC motor for wheels

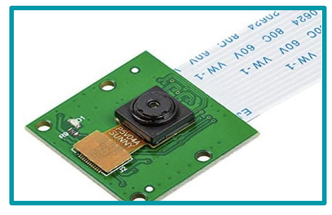


Figure 8 above: A photo of our replacement camera.

Conclusion & Next Steps

- Finishing the stoplight recognition AI and integrating motor driver.py with this program
- Train the AI on a large data set
  - Ensure that it behaves properly in practice
- Integrate pre-trained model with our own data set
- Test our model in a physical environment with our model stoplight

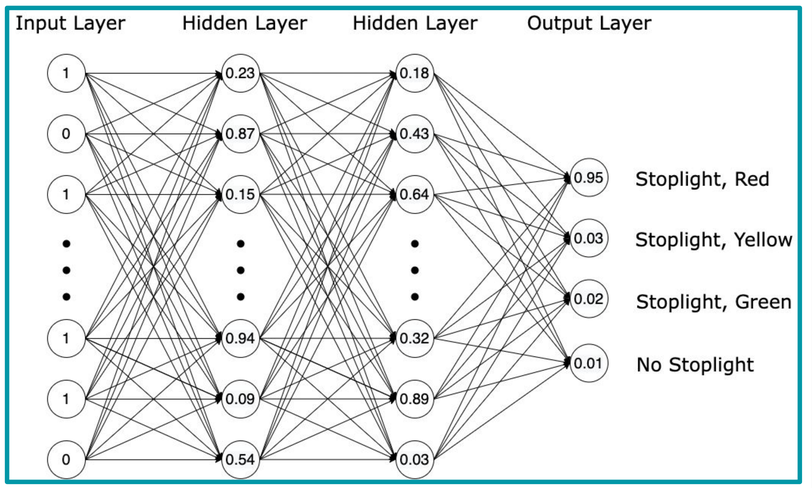


Figure 9: Diagram depicting how the neural network processes frames.