Mini Autonomous Car

Team4 (EGR 530)

Sudeeksha, Satya(TL), Dheeraj, Harsh, Jay, Raunakjit

Overview

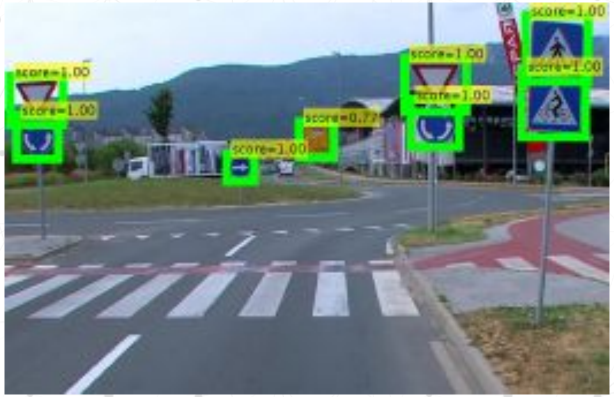
Introduction

- Car is a commodity to everyone in the US.
- Great revolution in whole car industry since 2-3 decades.
- Many types of cars with different features
- An observation is that to save time, the car industry is taking a path towards autonomous.
- Major autonomous car features expected
  - Detect obstacles
  - Drive smoothly
  - Understanding the real world scenarios and react.
- Is this easy?

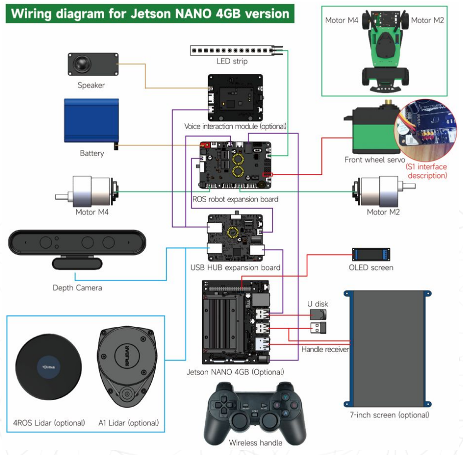
Problem Statement

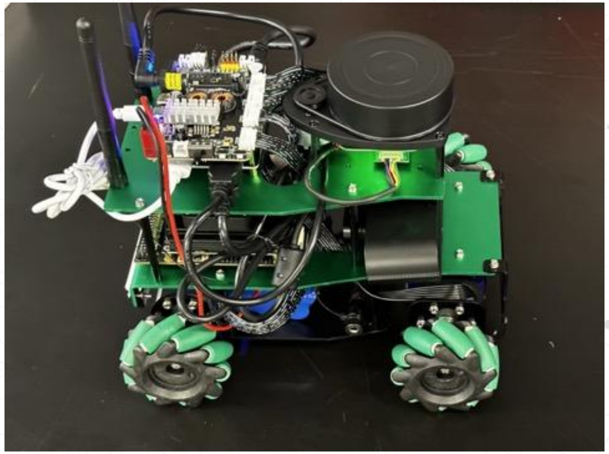
We chose to implement 2 features of an Autonomous car simultaneously which are

- Line follow
- Traffic Sign detection and response



Implementation

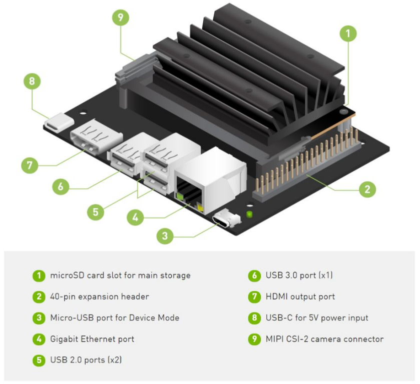


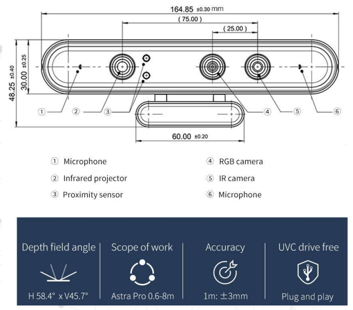


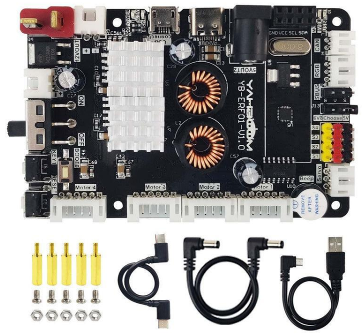
Implementation

Specifications

- Jetson Nano (4GB ram, 64 Gb memory)
- Astra Pro Camera
- Lidar
- 333 rpm motors
- Mecanum wheels


1 microSD card slot for main storage
2 40-pin expansion header
3 Micro-USB port for Device Mode
4 Gigabit Ethernet port
5 USB 2.0 ports (x2)
6 USB 3.0 port (x1)
7 HDMI output port
8 USB-C for 5V power input
9 MIPI CSI-2 camera connector





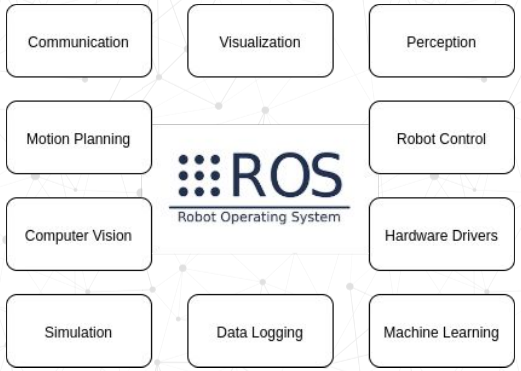
System Architecture

Hardware: Astra Camera, motors, motor driver, mecanum wheels, batteries.

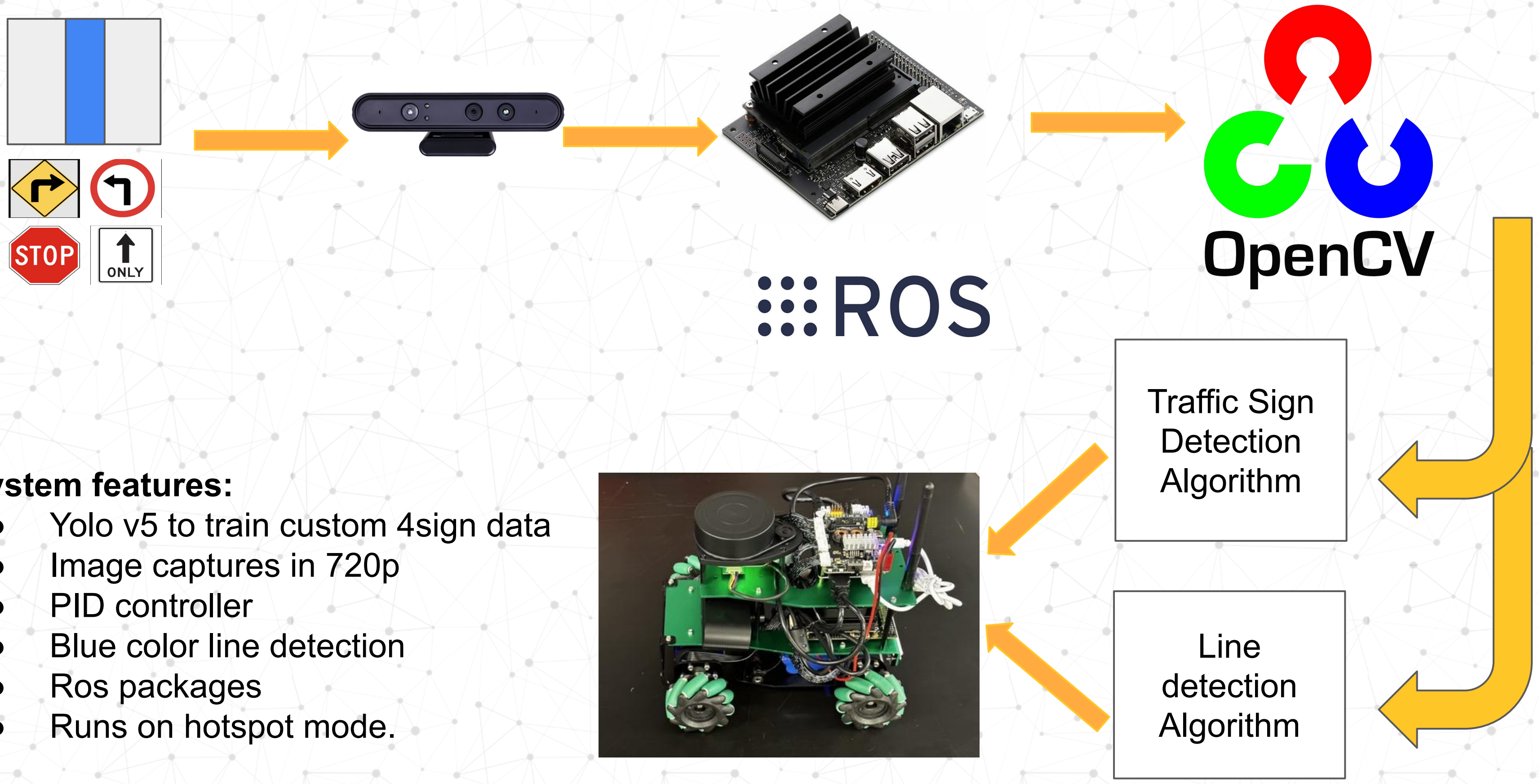
Software: ROS2, Python packages and libraries (Open CV), Linux

Input: Camera Image(with blue line, traffic signs)

Output: Velocities to motors, follow line, sign detection and response.



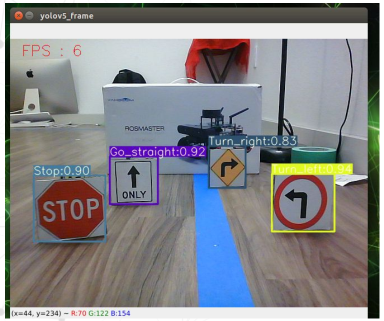
System Architecture

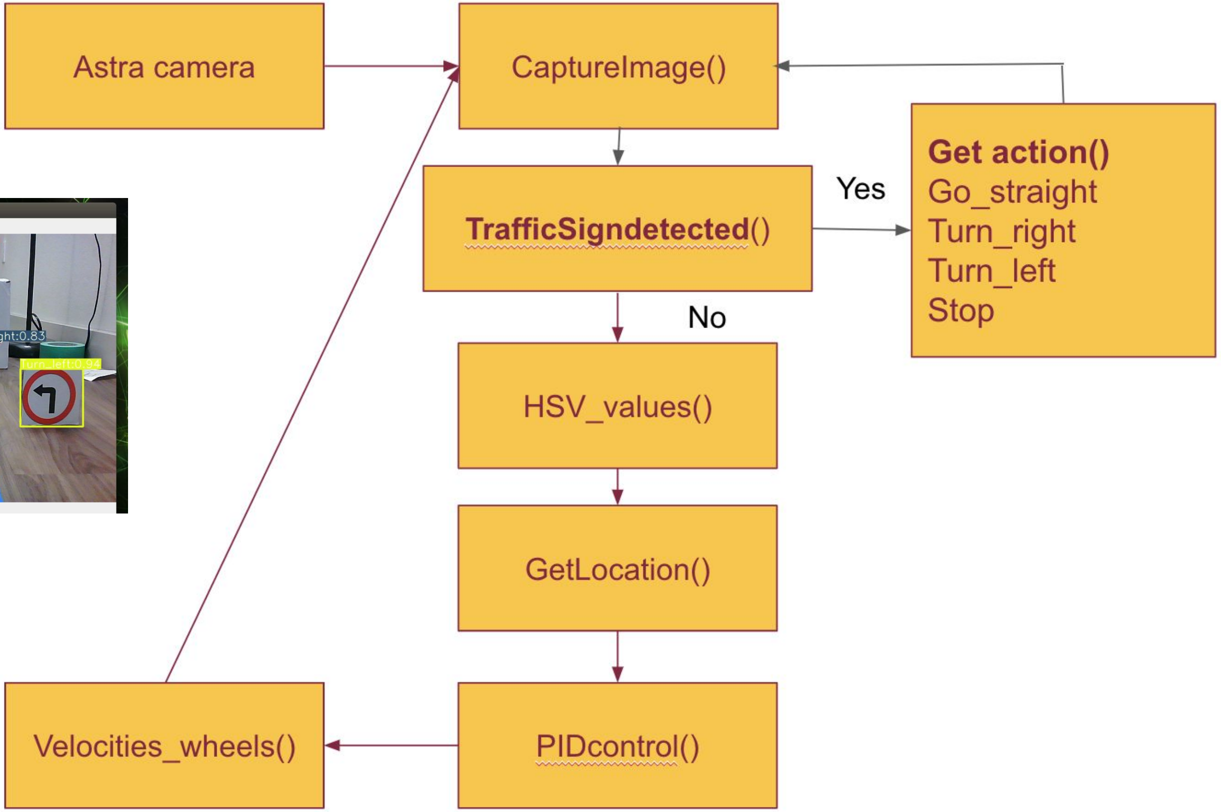


System features:

- Yolo v5 to train custom 4sign data
- Image captures in 720p
- PID controller
- Blue color line detection
- Ros packages
- Runs on hotspot mode.

Algorithm

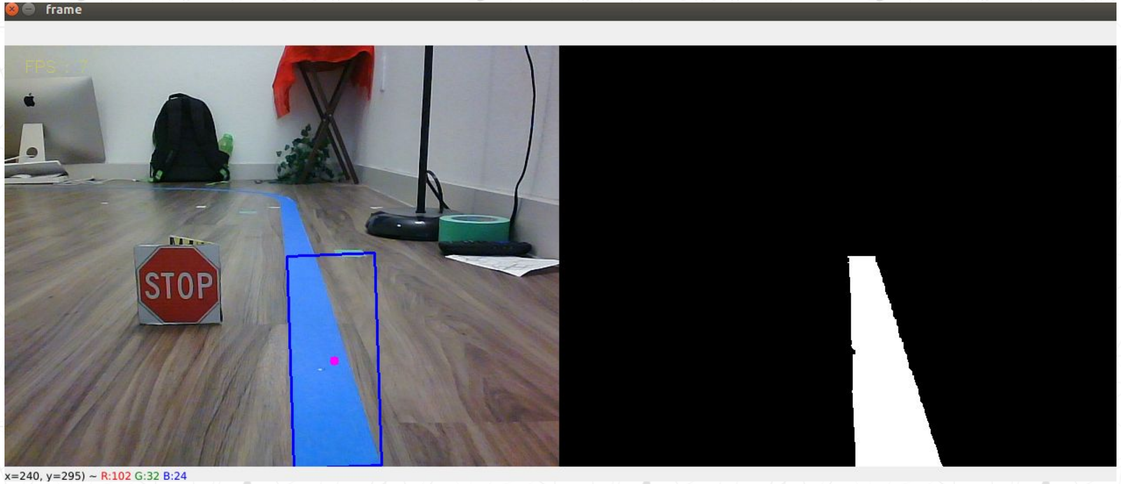

yolov5_frame
FPS : 6
ROSMASTER
Go_straight:0.92
Turn_right:0.83
Stop:0.90
Turn_left:0.94
STOP
ONLY
(x=44, y=234) ~ R:70 G:122 B:154


Astra camera
CaptureImage()
TrafficSigndetected()
Yes
No
Get action()
Go_straight
Turn_right
Turn_left
Stop
HSV_values()
GetLocation()
PIDcontrol()
Velocities_wheels()

Scope of Project

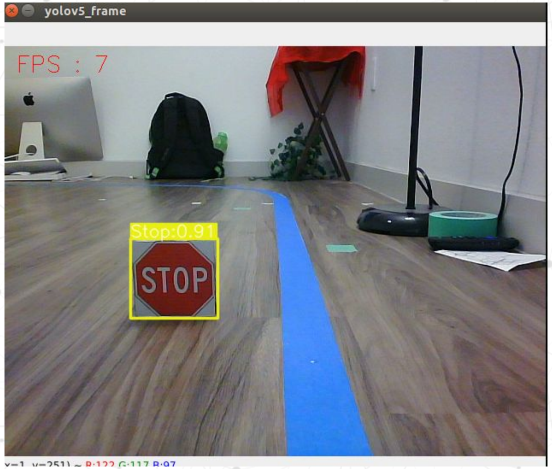
1. Any colored line following can be done but need to be calibrated.
2. 4 different sign detection is possible in our bot.
3. Responding to 2 signs: Go_Straight and Stop.
4. Response has a lag of 15-20 secs due to multiple processing..
5. Signs detection is about 80-95% accurate.
6. Detection of signs are at a distance of 15-20 inches.
7. No obstacles detection is done.
8. Inference is done on the bot itself.
9. Training is not recommended on bot.
10. 1 sign detection in the frame at a time.

Results



x=240, y=295) ~ R:102 G:32 B:24

Line detection



x=1, y=251) ~ R:122 G:117 B:97

Traffic Sign detection

Demo

https://youtu.be/VjgShMirpmE

Potential Challenges

- Lag in real time detection and responding.
- CPU overloading.
- Velocity value overlapping.
- Battery life.
- Correct calibration of color is required(environment specification is required).
- Camera angle.

Improvements

1. Implementation to take turns.
2. Obstacle detection and avoidance.
3. More real time detection with multiple objects.
4. Reduce lag by using GPUs.
5. Optimise algorithm to take decisions.
6. Increase the data set of traffic signs.
7. Implement automatic parking.
8. Detecting multiple traffic signs.
9. Take decision as per priority.
10. Making Calibration easier with respect to environment.

Thank you!

Special thanks to Dr. Junfeng Zhao