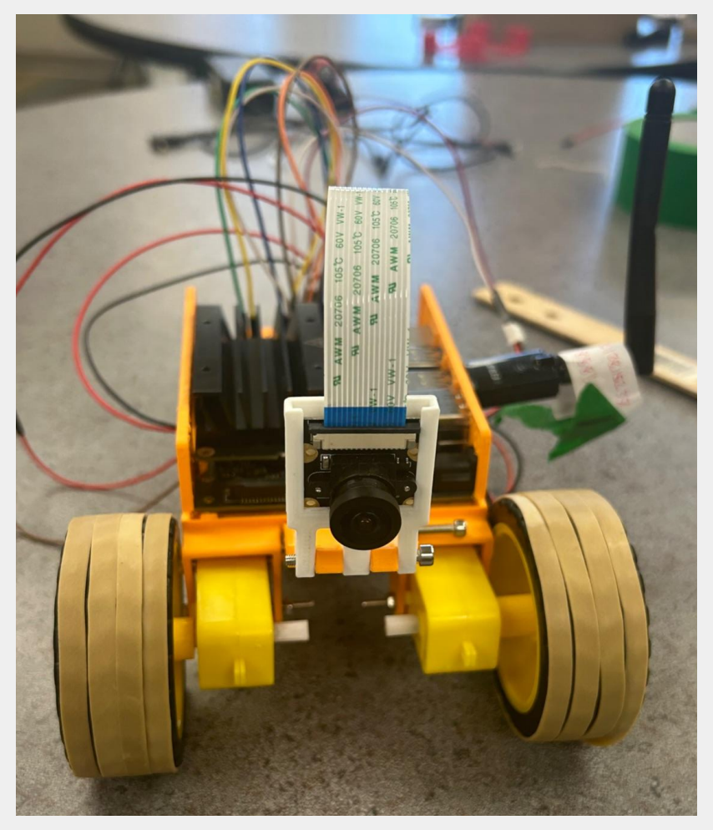
Amplifying Autonomous Driving with JetBot Vehicles

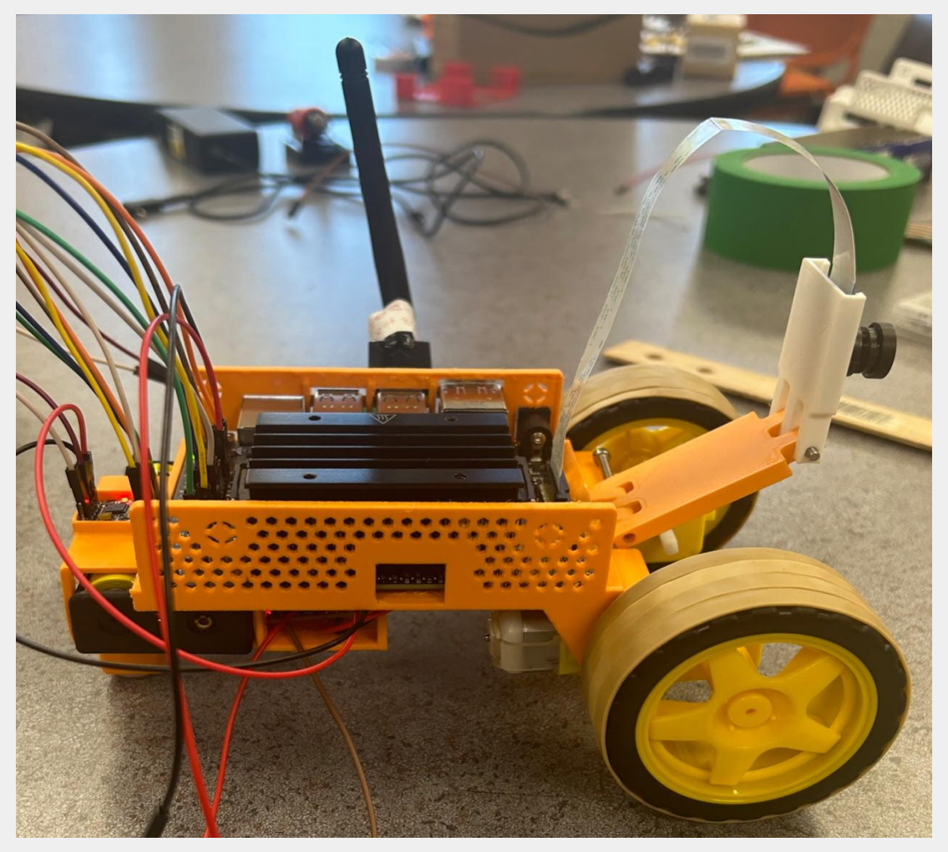
Students: Nicholas Barlow (Alabama State University), Yuki Lai (City University of Hong Kong), Franklin Zhang (University of Tennessee)
Mentors: Dr. Kwai Wong (University of Tennessee), Patrick Lau (University of Tennessee)

Introduction

Autonomous vehicles are key to increasing safety and reducing costs in driving and navigation. They also serve as an engaging example of artificial intelligence and robotics that can drive interest in the fields. However, they are relatively expensive and inaccessible.

This project tackles this with custom autonomous JetBot vehicles powered by NVIDIA Jetson Nano:

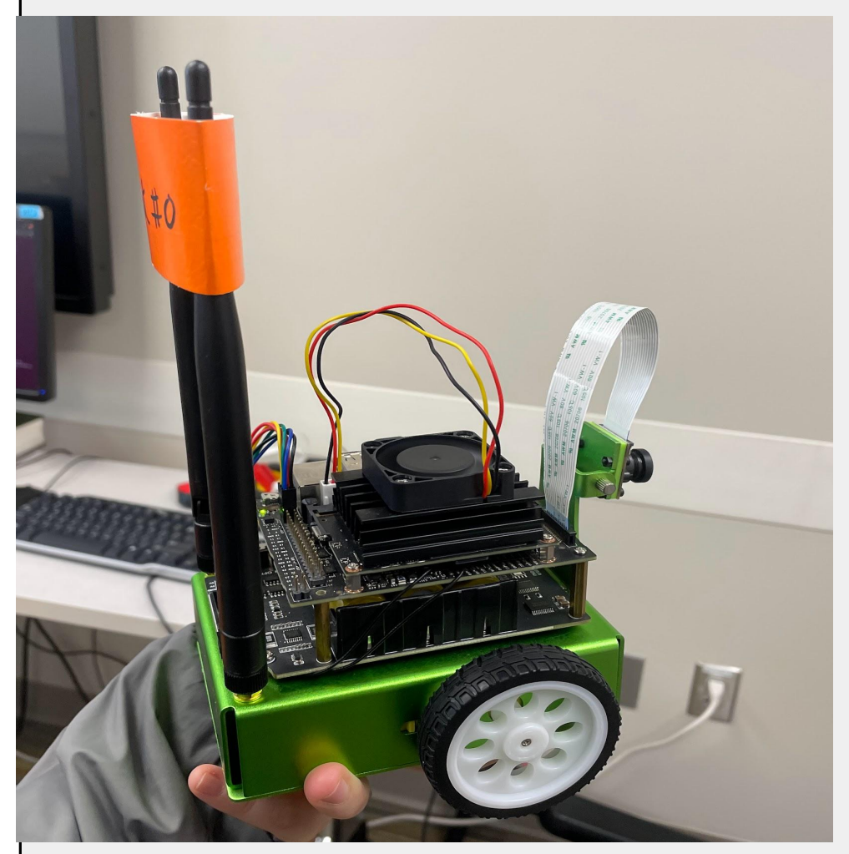




- Demonstrating applications in autonomous driving and navigation
- Computer vision AI techniques: classification, object detection
- Developing much less costly JetBots than commercial JetBot kits that are commonly used, increasing the accessibility of machine learning robotics explorations

The New JetBot

Commercial: $350

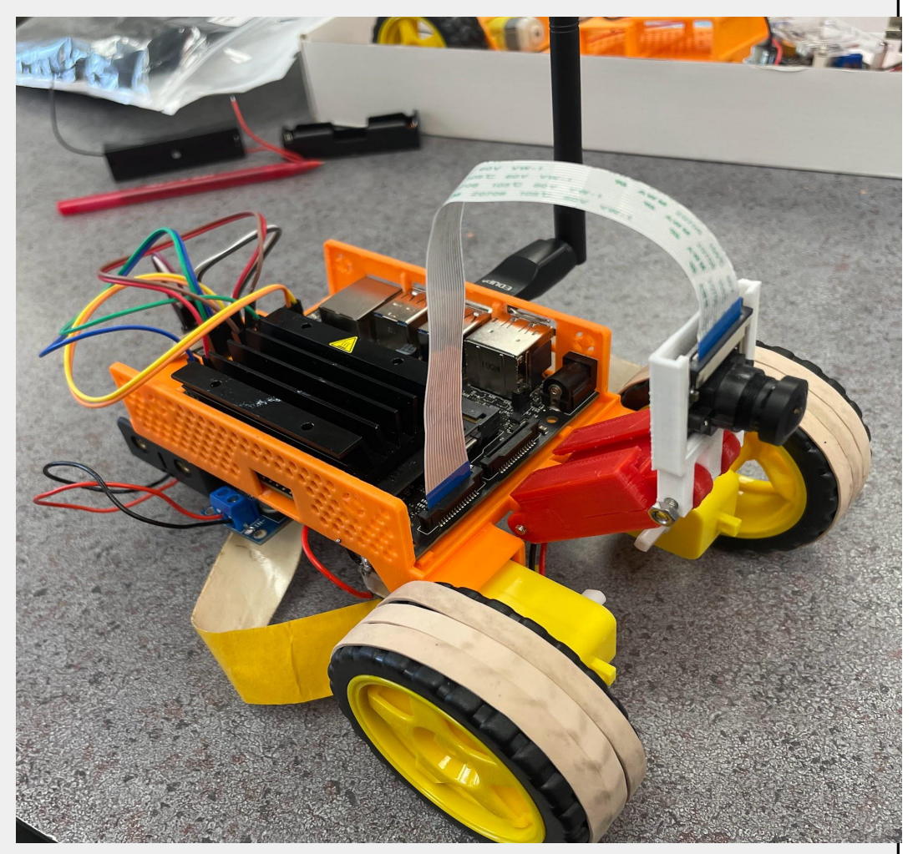


37% cheaper

Same autonomous driving capabilities

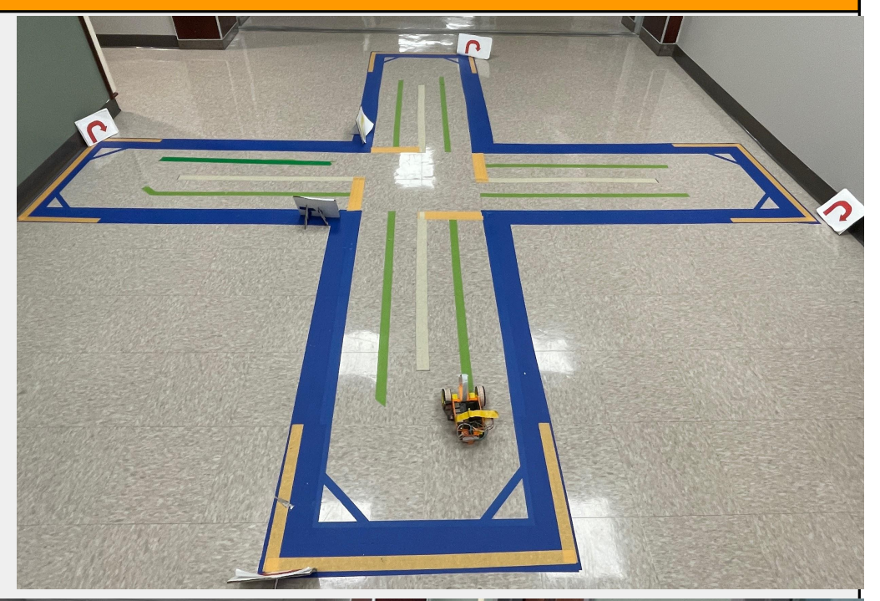
Clearly documented building steps

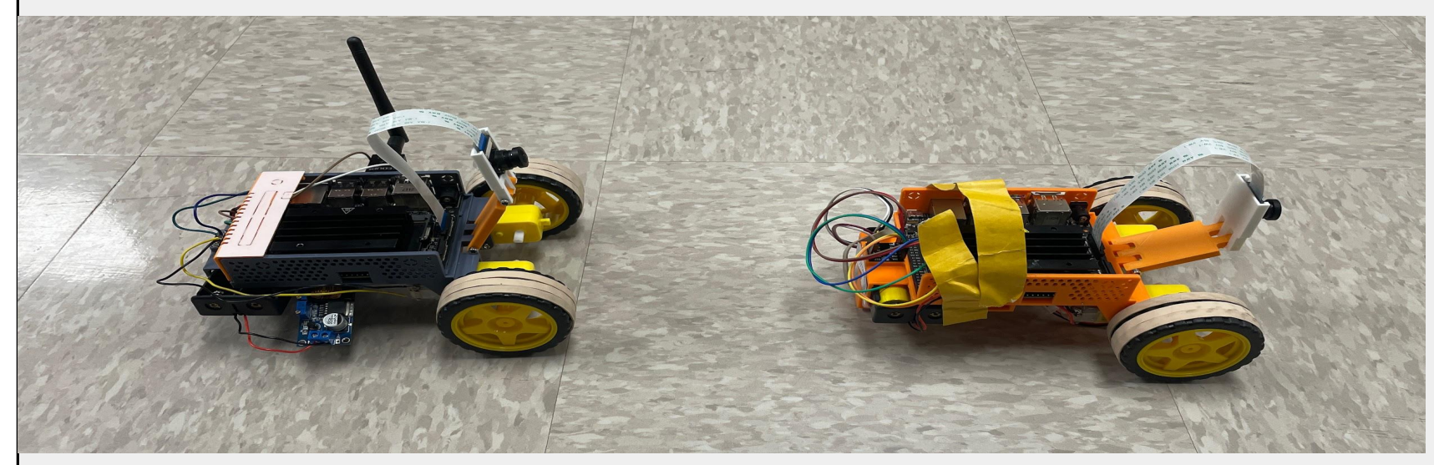
New: $220

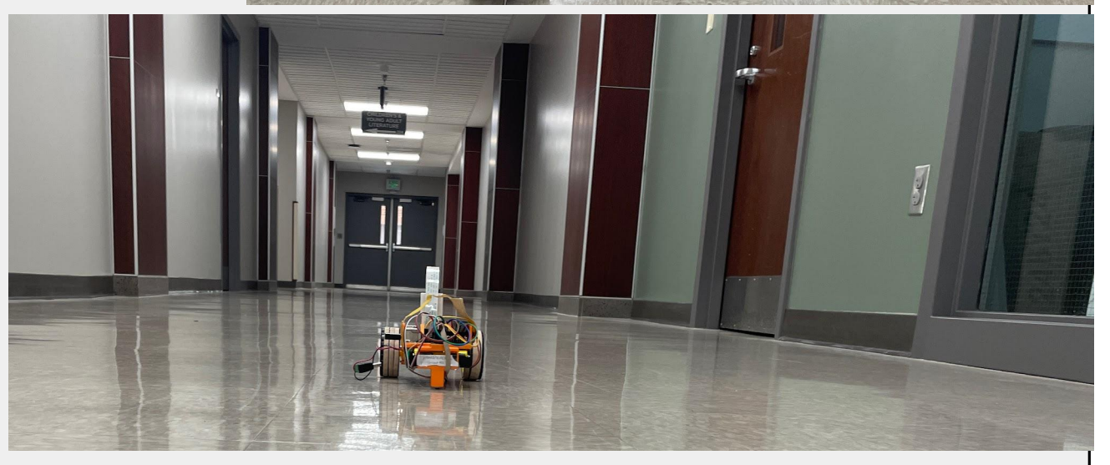


Computer Vision Tasks

- Navigating intersection (running on track): road following, sign detection
- Car following
- Corridor navigation
- Collision avoidance







Training Process

Basic process:

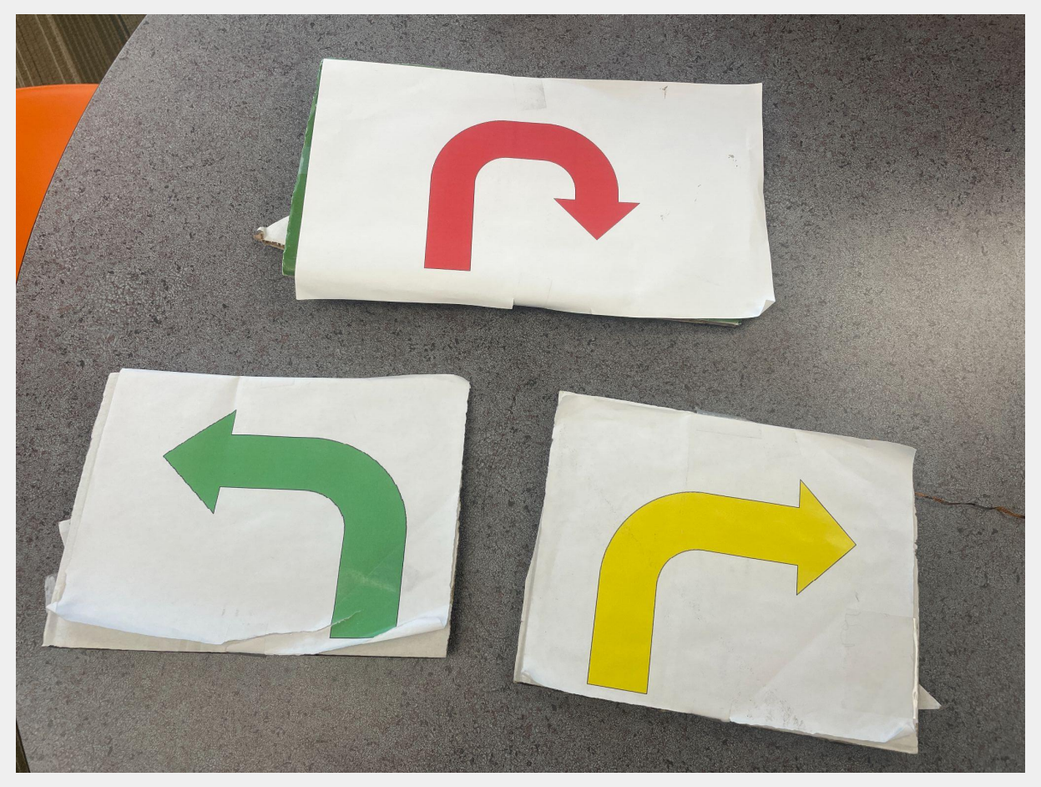
1. **Data collection** – Image capture (GStreamer)
2. **Annotation** – Computer Vision Annotation Tool (CVAT) bounding boxes for detection, or simple grouping of images for classification
3. **Training** – PyTorch code from Jetson-Inference
4. **Inference** – Original code powered by TensorRT

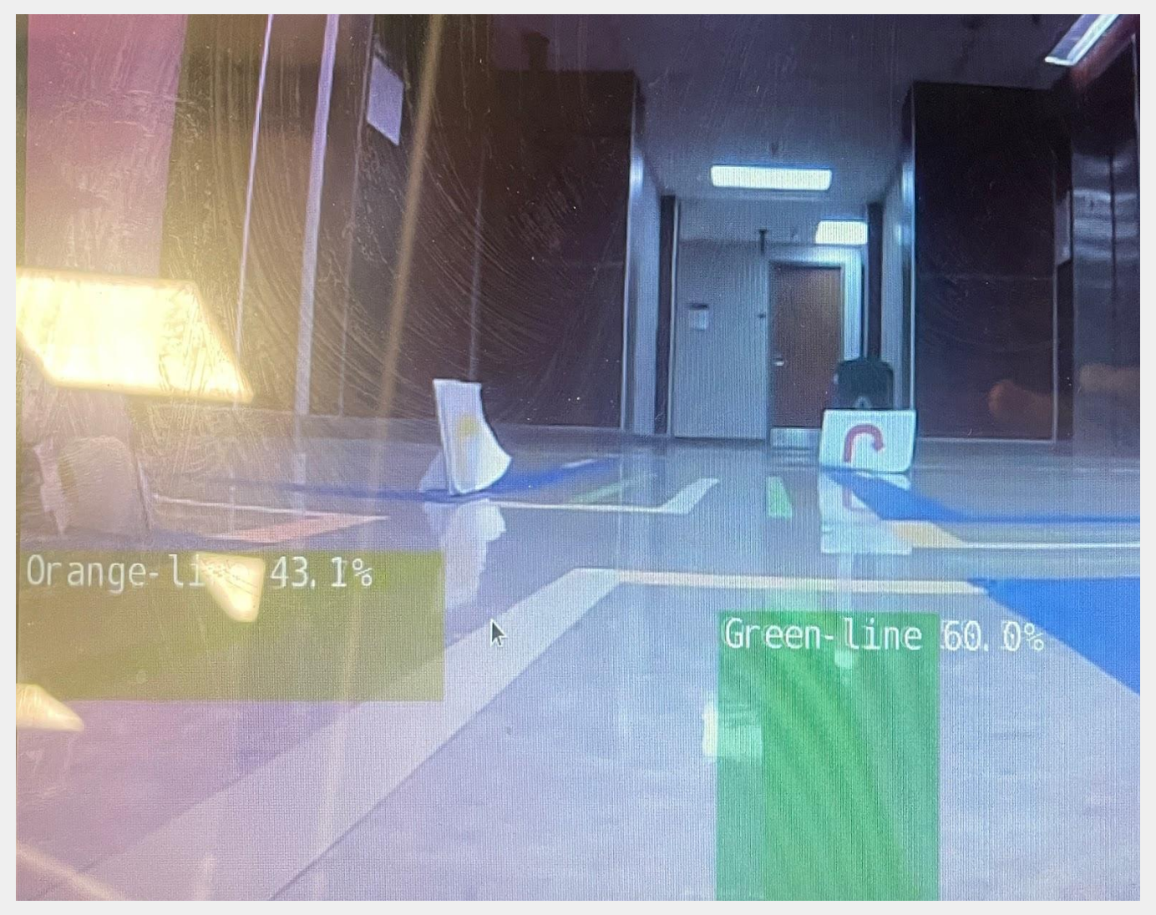
Tasks and Results

New JetBot **Setup:**

Goal: Fully compatible with the track (intersection) navigation program used in our previous JetBot model

- Accurately detect green and orange lines on the track, and follow the instruction based on the color of the line
- Accurately detecting and responding to road signs like left turns, right turns, and U-turns and executing specific actions for each sign and then seamlessly returning to following the road line

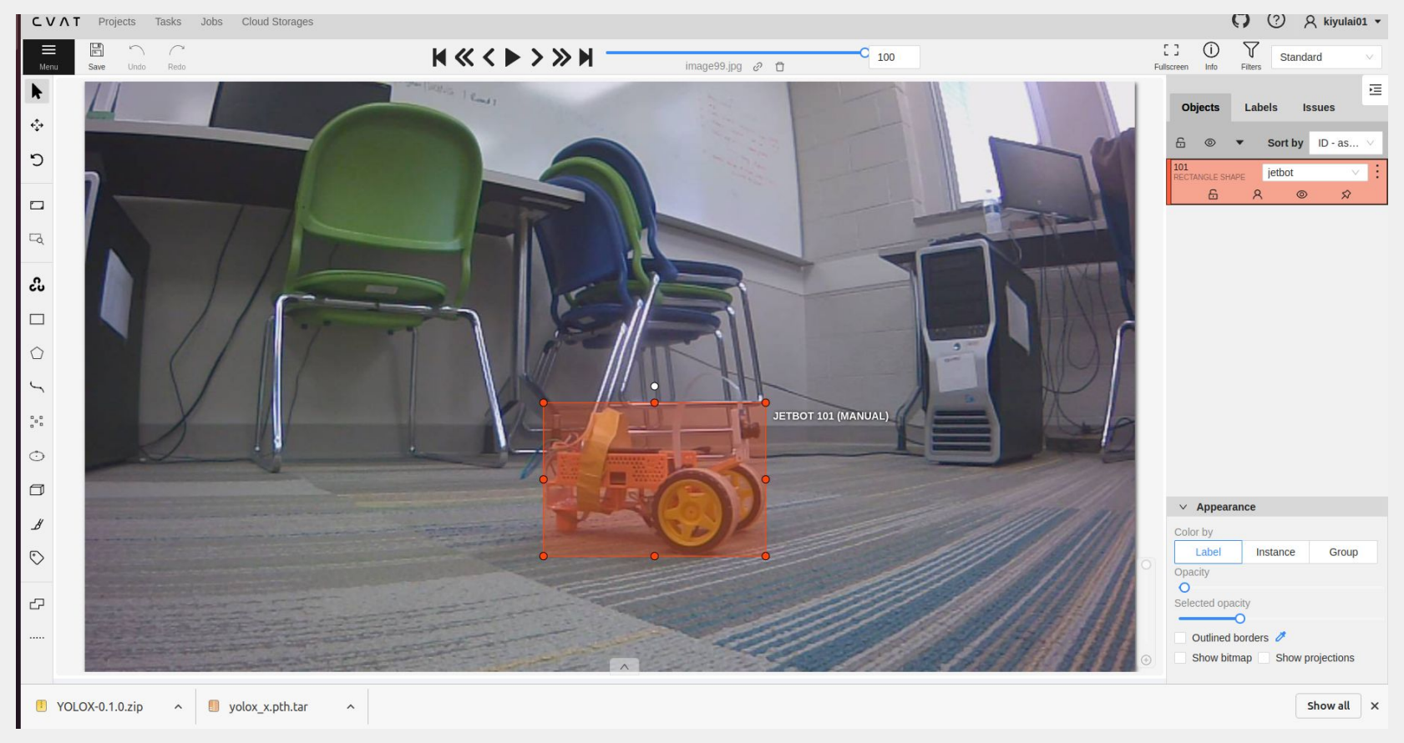




Car following:

Goal: Detect and follow another JetBot in front of the camera

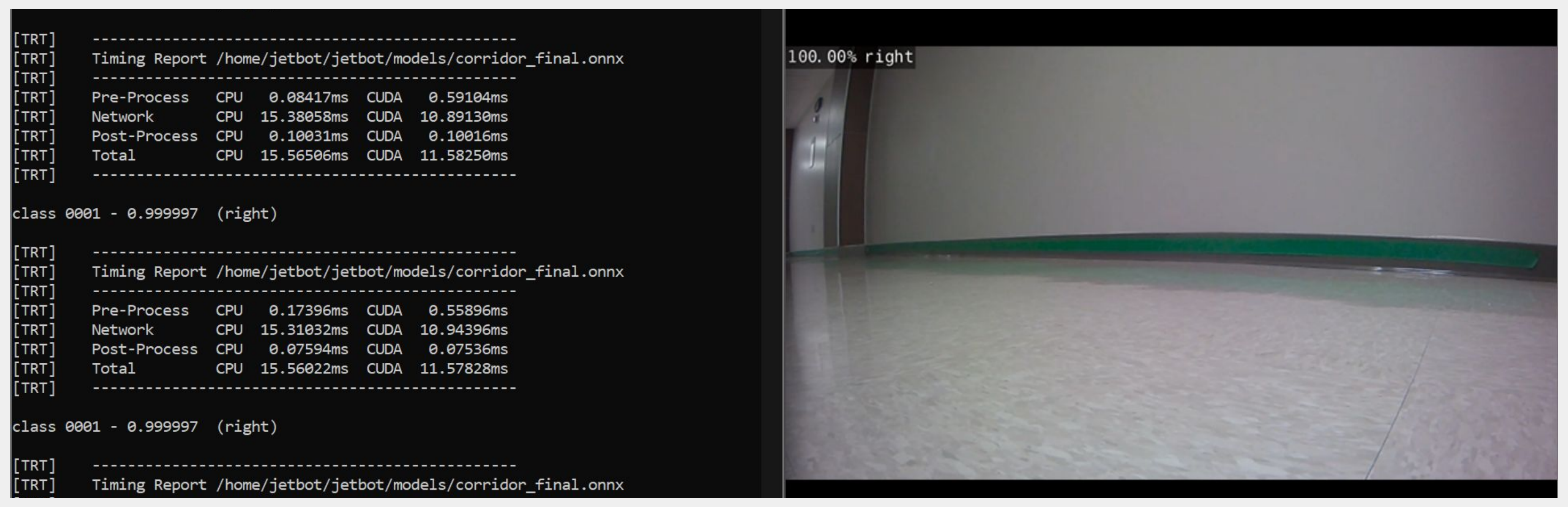
- Capture 100 images of a new orange car using the camera of the JetBot
- Label the images using CVAT
- Export as a visual object class dataset and train a object detection model using PyTorch code from Jetson-Inference
- Convert the saved PyTorch model to Open Neural Network Exchange (ONNX) model
- Develop a Python program that instructs the JetBot to move based on the position of another orange JetBot detected and shown in camera feed



Corridor navigation:

Goal: Drive a JetBot through a corridor without relying on line instructions

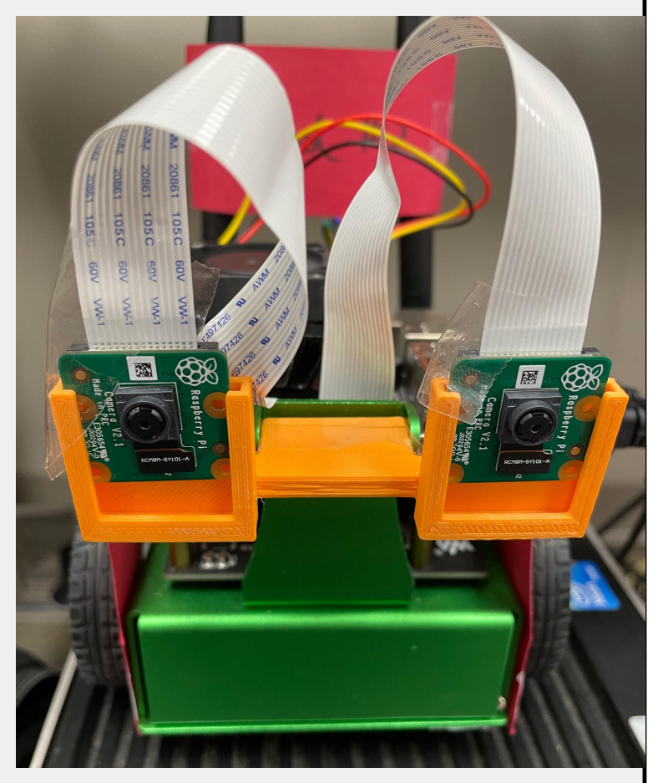
- Capture 3000 images of the left and right side of the wall and the road
- Organize the images into three folders with the names Left, Right, and Straight, and split them into training (80%), testing (10%), and validation (10%) sets to train a classification model
- Develop a Python program that enables JetBot to move based on the direction of the camera it is facing



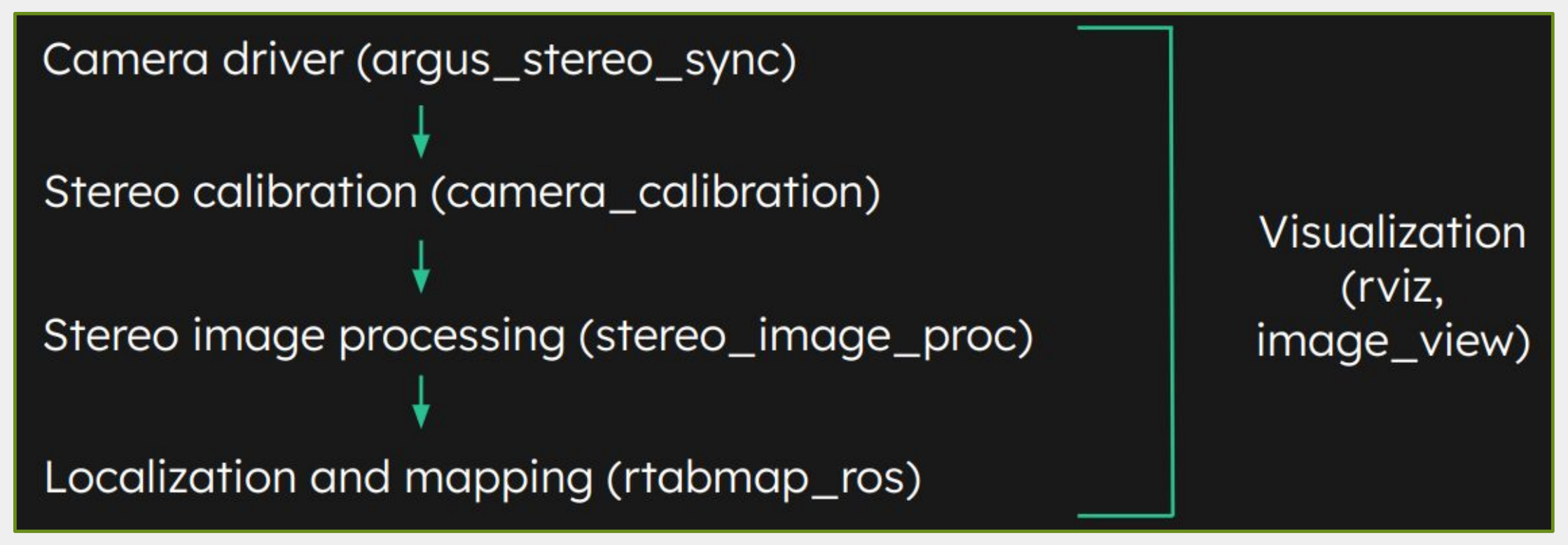
```
[TRT]    ----------------------------------------------
[TRT]    Timing Report /home/jetbot/jetbot/models/corridor_final.onnx
[TRT]    ----------------------------------------------
[TRT]    Pre-Process   CPU   0.08417ms  CUDA   0.59104ms
[TRT]    Network       CPU  15.38058ms  CUDA  10.89130ms
[TRT]    Post-Process  CPU   0.10031ms  CUDA   0.10016ms
[TRT]    Total         CPU  15.56506ms  CUDA  11.58250ms
[TRT]    ----------------------------------------------

class 0001 - 0.999997  (right)

[TRT]    ----------------------------------------------
[TRT]    Timing Report /home/jetbot/jetbot/models/corridor_final.onnx
[TRT]    ----------------------------------------------
[TRT]    Pre-Process   CPU   0.17396ms  CUDA   0.55896ms
[TRT]    Network       CPU  15.31032ms  CUDA  10.94396ms
[TRT]    Post-Process  CPU   0.07594ms  CUDA   0.07536ms
[TRT]    Total         CPU  15.56022ms  CUDA  11.57828ms
[TRT]    ----------------------------------------------

class 0001 - 0.999997  (right)

[TRT]    ----------------------------------------------
[TRT]    Timing Report /home/jetbot/jetbot/models/corridor_final.onnx
```

Upgrades with Stereo Vision

Goal: Apply **stereo vision** (dual camera) with ROS packages to enable SLAM (simultaneous localization and mapping) and navigation on the JetBot – e.g., navigating through hallways in a building to a desired destination.

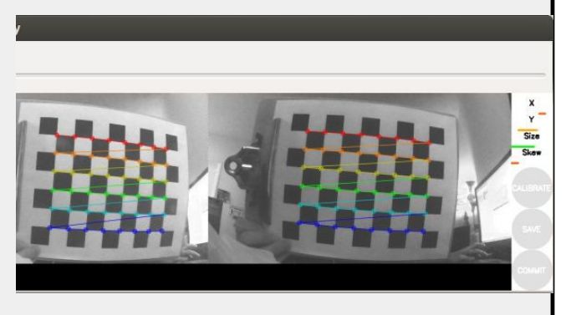


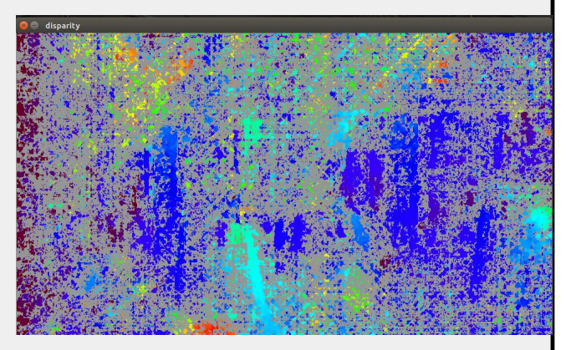
We have attempted this ROS package flow that generates a depth map and point cloud before attempting to use it for SLAM:



Camera driver (argus_stereo_sync)
↓
Stereo calibration (camera_calibration)
↓
Stereo image processing (stereo_image_proc)
↓
Localization and mapping (rtabmap_ros)

Visualization (rviz, image_view)



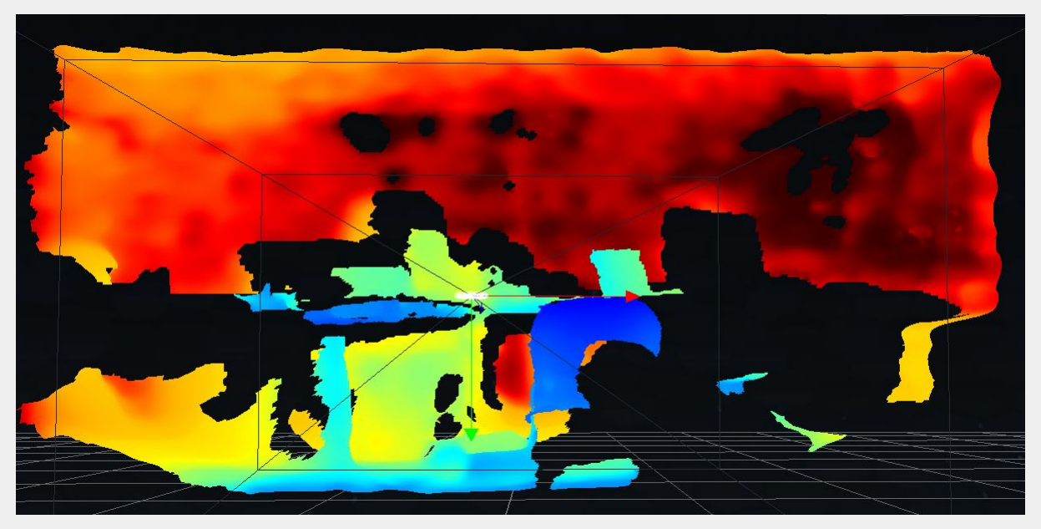


However, we have run into multiple issues:

- Camera synchronization requires a camera driver that, along with cheap cameras, leads to poor image quality
- The Jetson Nano regularly crashes in the process

We have moved to the more powerful Jetson Orin Nano to deal with the crashes. However, the first issue persists, so we have switched to a Realsense depth camera.

We have found much better results with stereo image processing, and we will continue to test the SLAM package and set up the ROS navigation stack.



Conclusion

We have successfully demonstrated several different autonomous driving tasks using AI computer vision techniques on the JetBot. We have also designed and constructed many inexpensive JetBots and documented the setup steps clearly, increasing the accessibility of robotics and AI, especially to educational programs that can increase interest in these fields.

In the future, we hope to further enhance the AI capabilities of these JetBots, spread our work to promote the accessibility and use of JetBots, and complete development navigation abilities with stereo vision on a higher-end version of the JetBot.

Acknowledgments

This project was sponsored by the National Science Foundation through Research Experience for Undergraduates (REU) award, with additional support from the Joint Institute of Computational Sciences at University of Tennessee Knoxville.

References

Franklin, Dustin. “Jetson-Inference.” *Dusty-Nv/Jetson-Inference.*, 18 June 2019, github.com/dusty-nv/jetson-inference.

Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, & Andrew Ng (2009). ROS: an open-source Robot Operating System. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*.

NVIDIA-AI-IOT, “NVIDIA-AI-IOT/jetbot: An educational AI robot based on Nvidia Jetson Nano.,” GitHub, https://github.com/NVIDIA-AI-IOT/jetbot (accessed Jul. 24, 2023).