AUTODRIVE CHALLENGE

Faculty Advisors: Dr. Steve Southward and Dr. Lynn Abbott

Yashit Agarwal , Antonio Alonso, Nate Doggett, Sammy Fellah, Rafiul Kabir, Josh Kruse, Yuxuan Li, May Schulte

PROBLEM STATEMENT

Virginia Tech AutoDrive Design Team is tasked to **develop and demonstrate a level 4 autonomous vehicle.** The technical goal of the competition is to **navigate an urban driving course** in an autonomous mode to simulate real life traffic scenarios encountered on public roads.

ULTIMATE GOAL: WIN THE COMPETITION

OUR ROLE

To design the **perception** and **path planning algorithms** to successfully compete in a series of challenges. Challenges include navigating intersections and highways, detecting obstacles and road signs, and adhering to safety requirements.

SYSTEM REQUIREMENTS

SIGNAGE

1. Detect and classify **traffic signs**
2. Read and report information on detected **traffic signs**

PAVEMENT MARKINGS

3. Detect and classify **lane lines** on a road
4. Determine **color** of detected lane
5. Determine **lateral position** of the vehicle within a lane
6. Detect and classify **stop lines** on a road
7. Detect and classify **crosswalk lines** on a road
8. Detect the **number of detects lanes**
9. Detect **which lane** the vehicle is in

RE-ROUTING

10. **Monitor inputs** from the perception algorithms to determines whether a global or local reroute is required
11. **Generate an alternate route** when the physical path is no longer viable
12. **Start from the current vehicle position** so that the passengers do not experience a discontinuous motion
13. **Generate an alternate route** when a speed change is required

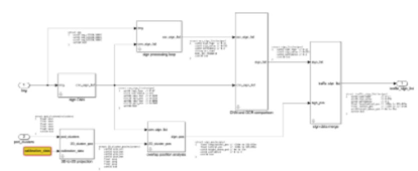
ACCURACY AND IMPLEMENTATION

14. Validated accuracy of at least **90% in normal conditions**
15. Validated accuracy of at least **60% during inclement conditions**
16. Report accuracy of **each detection**
17. Report accuracy of **each classification**
18; Implement on the **computational platform** provided by Intel
19. Produce all outputs within **100ms** of receiving an image

DESIGN

SIGNAGE

System Architecture:



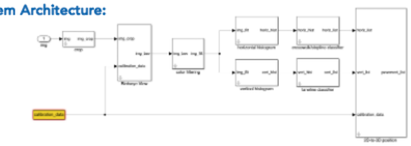
Our design primarily employs a combination of Convolutional Neural Networks (CNN) and Optical Character Recognition (OCR) to accurately classify traffic signs. To obtain the necessary image and data, we utilize camera and LIDAR technology to capture high-quality images and record the precise coordinates of detected objects. This data is then used to match and recognize the target object, providing both the recognition result and the corresponding coordinates.

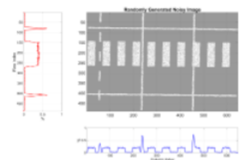
For instance, if an image captured by the vehicle indicates that there may be a traffic sign ahead, the image is first sent to the "sign CNN" for primary classification. Subsequently, the result and image are transmitted to the "sign processing loop" to gather additional results from OCR. Once both the CNN and OCR have generated results, the "CNN and OCR comparison" node compares them and assigns a confidence level to the recognized sign.

In addition to image data, the system also gathers point cloud data from LIDAR, which is transmitted to the "3D-2D projection" node to convert the 3D coordinates to 2D coordinates. This conversion helps to calculate the overlap rate in the next node, known as the "overlap position analysis. "In the given example, the blue portion indicates that the sign detection cluster has detected an object that might be a traffic sign and will subsequently send its coordinates to the "overlap position analysis" node for further comparison. If the coordinates from the cluster and the OCR match and refer to the same object, the 3D coordinates are transmitted to the next node for further processing.

PAVEMENT MARKINGS

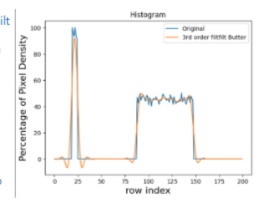
System Architecture:





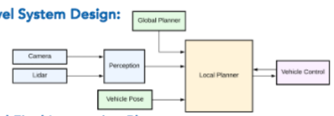
Row and Column Histogram:
The row and column histogram technique is a vital part of our pavement detection algorithm, allowing for accurate identification of lane lines, stop lines, and crosswalks. By analyzing the color distribution along the rows and columns of a randomly generated synthetic image, yellow and white pavement lines can be detected.

Noise Filtering Technique:
This noise filtering technique uses the inbuilt filtfilt method as part of the popular SciPy library. Noise filtering can help reduce high levels of noise in a randomly generated synthetic image, in effect, allowing us to analyze smoother histograms. With high level of noise, small fluctuations in pixel values are present. These fluctuations can make it difficult to accurately identify the relative and absolute peaks and valleys, which are critical for detecting and classifying lane lines. By applying a filter to the histogram, these fluctuations are be removed or reduced.

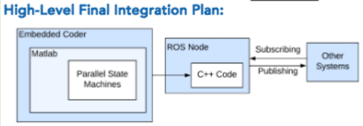


RE-ROUTING

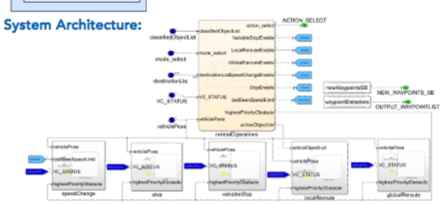
High-Level System Design:



High-Level Final Integration Plan:



System Architecture:



Speed Change FSM: Handles any necessary speed changes as the vehicle passes speed limit signs

Stop FSM: Handles when the vehicle identifies a stop sign. It ramps down the waypoint velocities and stops for a fixed amount of time (5 sec).

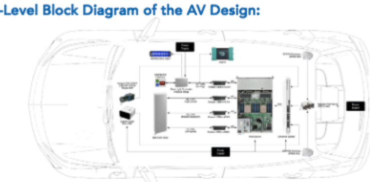
Variable Stop FSM: Handles any stops that are not a predefined amount of time. Examples include stop lights, pedestrians, animals, railroad crossings, etc.

Local Re-route FSM: Handles the scenario of going around an obstacle blocking the current path. Returns to the original path after passing the obstacle.

Global Re-route FSM: Handles the scenario where the current path is blocked, but the vehicle cannot perform a local re-route. Global Planner is called to create an entirely new route. If no route is available, the vehicle exits autonomously.

SYSTEMS SAFETY

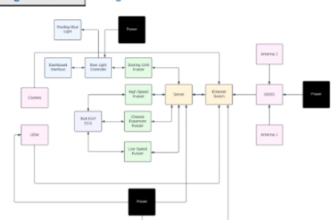
High-Level Block Diagram of the AV Design:



Hazard Identification	Hazard Description
PHA 1	Lidar Sensor Faults
PHA 2	Camera Sensor Faults
PHA 3	GPS Faults
PHA 4	IMU Faults
PHA 5	Wheel Encoder Faults
PHA 6	Communication Faults
PHA 7	Perception Faults
PHA 8	Route Planner Faults
PHA 9	Vehicle Control Faults
PHA 10	Power Faults
PHA 11	Blue Light Faults

Preliminary Hazard Analysis Table:
The first step in our safety analysis process was to perform a Preliminary Hazard Analysis (PHA), which analyzes every major function of the system. From the previous year, only 7 hazard categories were identified, and General Motors wanted the scope to be expanded. Therefore, in our analysis, we identified 4 more potential hazard categories, including the addition of the safety blue light required for Year 2 of the competition. As a result, 112 potential hazards were identified from these categories.

Functional Interface Analysis Diagram:
This diagram shows every major hardware component and each interface connection. We started from this diagram to identify all potential safety hazards that occur in the same from an interface perspective. After our analysis, we identified 18 System Safety Requirements to ensure the interface connections are secure



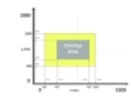
VALIDATION RESULTS

SIGNAGE

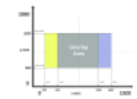
Outputs:

['limit', 'speed']

['turn', 'left', 'no']



[The overlap rate is 1, they are same object]



[The overlap rate is 0.9, they are same object]

['stop']

['turn', 'no', 'on', 'red']

PAVEMENT MARKING

Description	Color	Noise	Percent Accuracy w/ Respect to Ground Truth
Lane lines on a road	W/Y	Y/N	~100% for 0-20% noise over 1000+ iterations ~50% for 25%-35% noise over 1000+ iterations
The color of detected lane lines	W/Y	Y/N	~100% accuracy
The number of detected lanes	N/A	N/A	~100% accuracy
Lateral position of the vehicle within a lane	N/A	N/A	~100% accuracy
Which lane the vehicle is in	N/A	N/A	~100% accuracy
Crosswalks on a road	W	Y/N	~100% accuracy
Stop lanes on a road	W	Y/N	~100% accuracy
Crosswalk pixel index error	N/A	N/A	~100% for 0-20% noise over 1000+ iterations ~98% for 20%-40% noise over 100+ iterations
Stop line pixel row index error %	N/A	N/A	~3% index error
Longitudinal position of the crosswalk from the vehicle	N/A	N/A	~97% for iterations over 1000+
Longitudinal position of the stop line from the vehicle	N/A	N/A	~99% for iterations over 1000+

RE-ROUTING

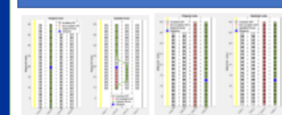


Figure 1.
Testbench for visualizing lane changes when an obstacle is in the current path. New waypoints are generated to avoid and return to the original route.

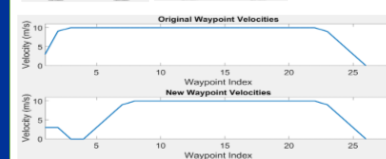


Figure 2.
After identifying a stop sign at waypoint index 3, the velocity waypoints show a gradual decrease in speed until a complete stop is reached

FUTURE WORK

For signage, we need to get the code modules to integrate with ROS, use ROS subscriber and publisher to receive and send ROS messages within the server, then combine all the outputs obtained from each node, package them, and publish it to the next node.

For pavement markings, we plan to use clustering, classification, and machine learning algorithms to automatically identify peaks in histograms and classify lane lines. By training with a machine learning algorithm like this on a large dataset of annotated images with labeled lane lines and corresponding peaks, the algorithms can learn to accurately identify the peaks in the histogram that correspond to the lane lines in a more robust manner.

For re-routing, we plan to do more robust testing of scenarios in simulation and the actual car.

For systems safety, we plan to continue testing and enact any mitigation strategies if hazards occur.