UW EcoCAR 3
ADAS: Advanced Driver Assistance Systems

Background
The goal of ADAS is to provide autopilot features to assist the driver and improve overall efficiency and safety.

This project involves creating optimized and accurate algorithms for object detection using concepts from the fields:
- Computer vision
- Machine learning/statistics
- Artificial intelligence
- Neural networks/deep learning

There is also a display and app aspect that requires knowledge in the following fields:
- Frontend and app development
- Data communication
- User experience
- Design
- Backend

Figure 1. Graphic showing lane detection capability. Matlab and computer vision techniques like sobel filter and Hough transform were used.

Driver Assist Features
EcoCAR 3 added the ADAS requirement (advanced driver assist system) which requires teams to implement some sort of system to assist the driver.

To achieve this, UW EcoCAR takes sensor data input in the form of range, relative velocity, and in-lane flags to determine when to begin braking for maximized regenerative braking.

Notifications and warnings of when to begin braking appear on a tablet. The system is transferrable to any android or iOS based system.

Sensor Fusion
The UW EcoCAR team uses two main sensors, a radar and Camera, and utilizes MathWork’s Automated Driving Toolbox to keep track of multiple objects across frames.

An extended Kalman Filter accurately tracks objects across multiple frames and updates data (such as range, range-rate).

UW EcoCAR offers undergraduate and graduate students alike the chance to work on a real-time Linux OS system, with development being in C++. Graduates have the opportunity to test out new models, and undergraduates have the chance to work among a team of engineers to develop software.

Figure 2. Regenerative Braking maximization system. Driver is within a certain distance, so the arrow indicates the braking roughness will be a medium harshness.

Figure 3. Camera Circuit Board and Delphi ESR Radar Sensors

Figure 4. UW EcoCAR ADAS System

Technology
Software
- C/C++: Machine learning, server and processing
- Matlab: Automated Driving Toolbox
- Python: Polynomial Fitting for Lane Lines
- NVIDIA Jetpack: Software tools for Jetson board
- React.js and Electron: Display and front end
- GitHub: Revision Control

Hardware
- 2016 Camaro: For demonstration
- Raspberry Pi: Web-based Server
- NVIDIA Jetson TX1: Computer vision, Neural net
- NXP S32V board: Computer vision
- Samsung display tablets: Display, UX testing

VIP Vertically Integrated Projects

UNIVERSITY of WASHINGTON