Roni Herschmann

ML pipeline for Collision Avoidance in Autonomous Vehicles

Focus on camera-based input for computer vision, with sensor fusion with radar or LiDAR, prioritizing real-time performance on embedded hardware like NVIDIA Jetson, and utilizing PyTorch and TensorFlow.

Pipeline Outline

1. Input processing: resize and normalize images from vehicle camera
2. Perception: object detection, segment drivable area, identify lanes
3. Hazard prediction/path planning: outputs from perception to predict trajectories and plan collision-free paths
4. Decision making and control: acceleration, braking, steering outputs
5. Fine-tuning: use pretrained models on datasets: BDD100K, nuScenes, Waymo Open

1. Perception
   - YOLOP: A multitask model for object detection, drivable area segmentation, and lane detection in a single pass. Real-time on Jetson TX2; pretrained weights are available. Pipeline: Feed frames for bounding boxes, drivable regions, and lane lines to flag path hazards. Integrates via PyTorch.
   - NVIDIA SegFormer: This offers semantic segmentation of road scenes, using the SegFormer-b5 model from HF, fine-tuned on Cityscapes at 1024x1024 resolution. This technology labels pixels as road, sidewalk, vehicles, and more, which assists in hazard identification. It can be integrated via Hugging Face transformers. Integrating this with YOLO would create a comprehensive scene map to filter for false positives.
2. Hazard Prediction/Path Planning
   - CarDreamer: The CarDreamer framework from GitHub, such as DreamerV3, simulates and learn driver behaviors.
   - Pre-trained checkpoints are available on Hugging Face and trained in the CARLA simulator for tasks such as overtaking other cars.
   - This is a step toward addressing scenarios that are deemed *unpredictable*.
3. Direct image to Action Mapping
   - NVIDIA DAVE-2: Convolutional neural network that maps raw camera pixels directly to steering commands, pretrained on roughly 72 hours of driving data across diverse driving conditions.
   - Architecture:
     - Normalization
     - 5 convolutional layers
     - 3 fully-connected layers
     - Outputs an inverse turning radius
4. Sources

- GitHubs: hustvl/YOLOP; ucd-dare/CarDreamer; danijar/dreamerv3; chingisooinar/End-to-End-Deep-Learning-Approach-for-Autonomous-Driving-DuckieTown.
- HF/Other: nvidia/segformer-b5-finetuned-cityscapes-1024-1024; NVIDIA DAVE-2 Blog.
- Academic Papers (See Q2 Writeup): Survey of Autonomous Vehicles' Collision Avoidance Algorithms; Vision-based Driver Assistance Systems: Survey, Taxonomy and Advances; Machine Vision-Based Autonomous Road Hazard Avoidance System for Self-Driving Vehicles.