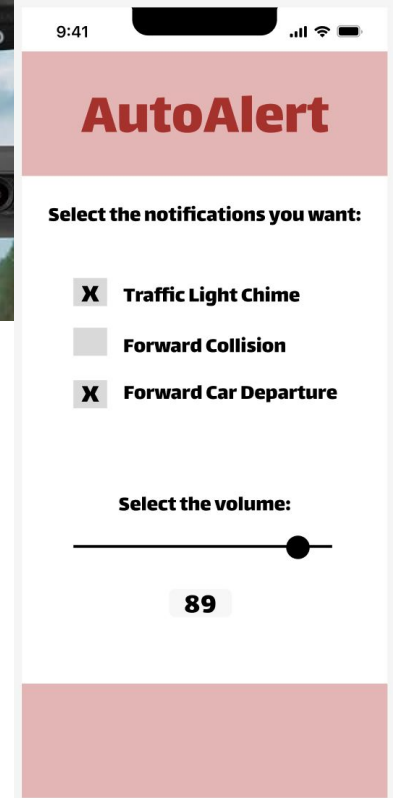
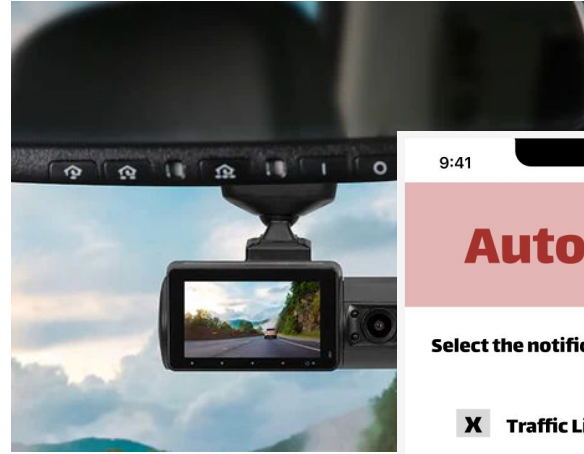


# Use Case: Problem + Project Intro

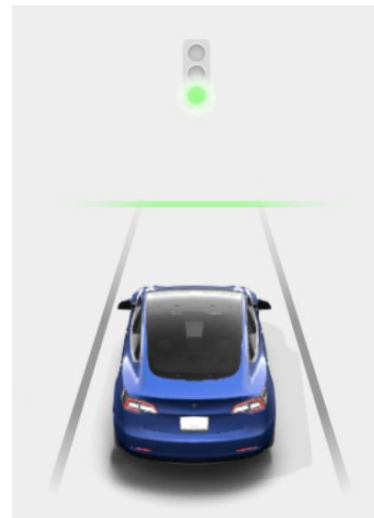
- Non-luxury cars and older cars don't have access to high-end safety features
- Bridge gap with affordable solution via dash cam that detects traffic lights, lanes, and obstacles that connects to mobile app that alerts driver



## Use Case: Application + ECE Areas

1. Light chime alert
2. Lane detection
3. Forward collision alert
4. Forward car departure alert

Implemented with software + hardware



# Use Case Requirements (1)

## Light Chime Alert

- Correct traffic light accuracy (multi-lane situation, different kinds of lights): **~90%**
- Notification sounds within **2s** after turning green
- Tesla: 1s

## Lane Detection

- Able to detect lane position up to **99%** accuracy
- Road quality
  - Bad (Construction, Closed Lanes, Faded/No Lines): **90%**
  - Good: **99%**

## Use Case Requirements (2)

### Forward Collision Alert

- Up to **50m** in front
- Active above **3 MPH** up to **40 MPH**
- Alert up to **99%** accuracy

### Forward Car Departure Alert

- Notification sounds when car in front moves **10ft+**
- Other luxury car: detect car that moves 10ft+ when stationary

# Technical Challenges (1)

## Light Chime Alert

- Unable to detect traffic light position / type of traffic light (specific to right-turn only, etc)
- Risk Mitigation: Modify nearest-light algorithm

## Lane Detection

- Unable to correctly identify left and right lanes
- Risk Mitigation: Reviewing openCV code, improving video processing code

## Technical Challenges (2)

### Forward Collision Detection

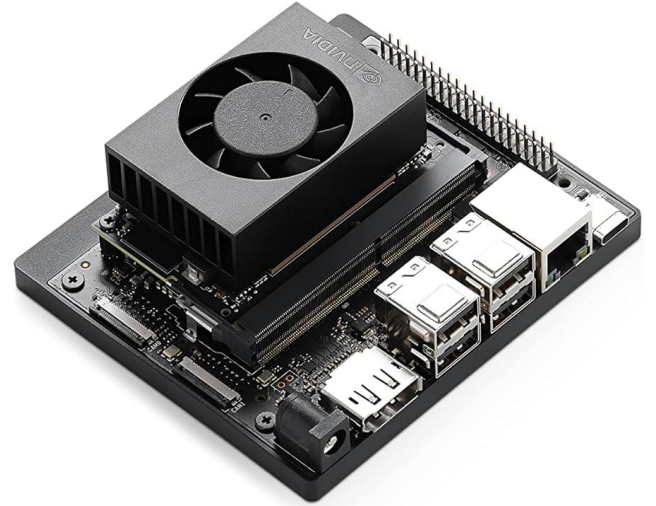
- Alert doesn't go off on time, driver crashes into car
- Alert goes off too frequently
- Risk Mitigation: Chip allocation – have one chip dedicated to forward collision

### Leaving Car Detection

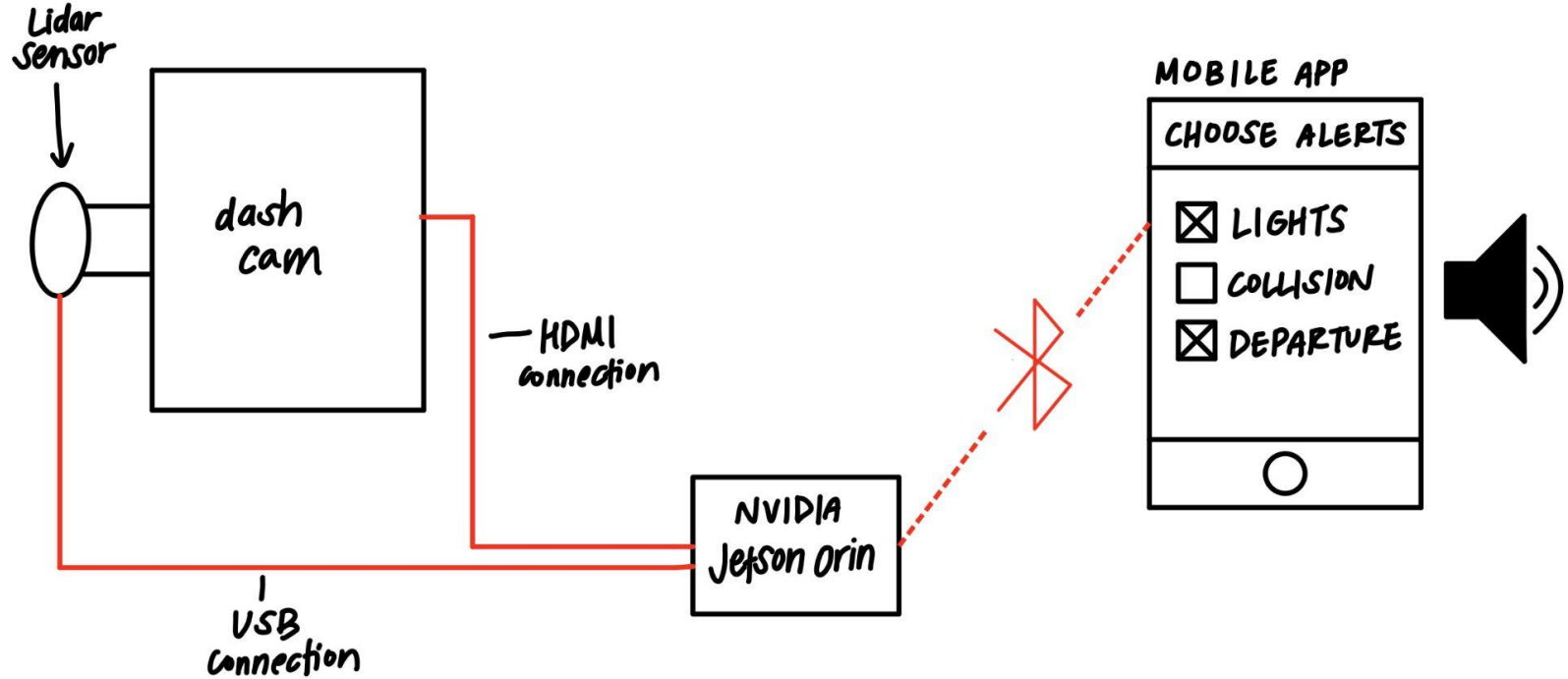
- Unable to detect car
- Car detection goes off when not necessary
- Risk Mitigation: Chip allocation – add more sensors to detect cars

# Solution Approach

- Hardware: NVIDIA Jetson Orin
- Software: Mobile development (Swift), object detection (OpenCV)
  - Mobile app: notification sound + selecting what notifications user wants
- Other:
  - Dash Cam
  - TeraRanger Evo 60 Lidar Sensor
  - Power source for hardware + dashcam
  - Wire connections (USB, HDMI)



# Solution Approach: Diagram





# Testing, Verification, and Metrics (1)

## Light Chime Alert

- Drive around with model running to see whether or not light changes are properly detected
- Measure how long notification takes
- % of accuracy (how many lights the alert is successful on)

## Lane Detection

- Drive around with model running and see whether or not adjacent lanes are properly detected

## Forward Collision Detection

- Use two toy cars and mount system to cars to safely simulate and test this since we can't do this in real life

## Leaving Car Detection

- Drive around with model running and see whether or not leaving cars are properly detected
- Measure how long notification takes
- % of accuracy (how many lights the alert is successful on)

# Testing, Verification, and Metrics (2)

## Demo

- Building track indoors
- 2 RC cars, 1 that holds system of dash cam and “box” and the other to simulate interactions between two cars



# Division of Tasks and Labor

## Ankit

Lane detection

Connections between camera, hardware, and software

## Emily

Forward collision detection

Forward car departure detection

## Eunice

Equipment procurement

Traffic light detection

Mobile application development

