CarMa

Team B7: Jananni Rathnagiri, Adriana Martinez, Evann Wu

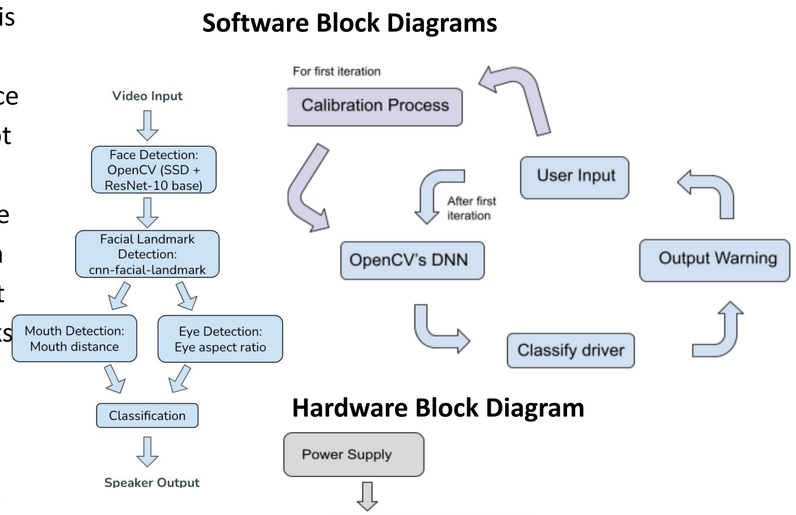
Product Pitch

CarMa is a driving assistant tool designed to guide drivers to follow safe protocols. CarMa is primarily used to help new and inexperienced drivers get comfortable on the road. Our driving assistant would simulate the role of a parent or driving instructor in the front seat warning the driver about distracted driving. The device has a camera pointed at the driver and using computer vision to detect if the driver is falling asleep or they appear to be distracted. CarMa focuses on ensuring the driver is focused on the road using computer vision to track the driver's face.

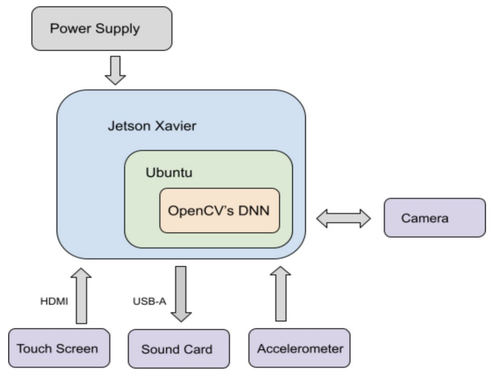
System Architecture

Each frame of the video capture is used for face detection using OpenCV and deep learning for face detector based on the Single Shot Detector (SSD) framework with a ResNet-10 base network. Then we obtain the facial landmarks which are used to localize and represent regions of the face. The landmarks are used for eye and mouth detection. We use both of these detections into a classifier to tell us if the user is distracted or not.

Software Block Diagrams



Hardware Block Diagram

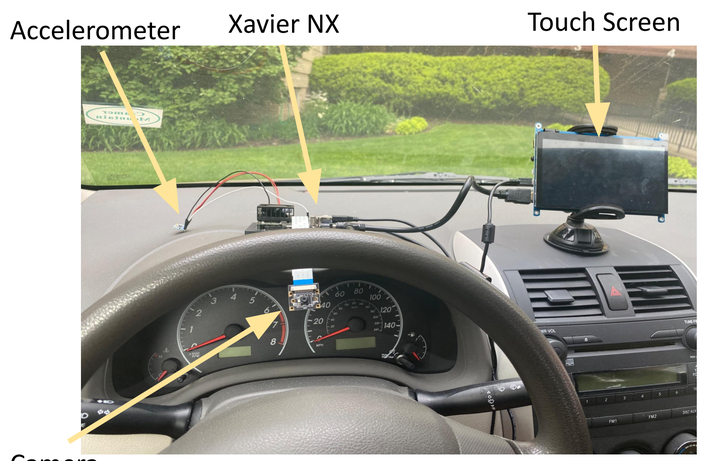


Our hardware diagram is centered around the Jetson Xavier NX. We chose to go with the Xavier NX due to it having the fastest CPU and GPU in our given price range. This allows us to have smooth real time image processing. For I/O, we use an accelerometer, a touch screen, and a camera for input. For output, we use the sound card.

System Description

For our system's input, we use a touch screen as the main user interface allowing drivers to start and stop the driver monitoring as well as calibrate the system, an accelerometer to enable our system when it detects the car is past 55 mph, which is what we have decided to be the scope of this project, and a camera for video input. We use a high pitched sound as output to alert the user when they are distracted or drowsy.

System Components



System Evaluation

In order to test our system, we created a suite of test videos simulating the actions of a distracted/sleepy driver and a focused driver. We collected videos from friends and family then kept track of the number of error events that occurred whether false positive or negative. To find accuracy, we divided the error rates by the total number of events for each of our individual tests.

Here are the error events we looked for:

- Eyes Closed for over 2 secs with no alert
- Blinking with an alert
- Yawns with no alert
- Talking with an alert
- Looking away for over 2 secs with no alert

Here is how our device did against our original project requirements.

Metrics, Validation & Testing

Requirements	Metrics	Test Plan	Test Results	Status
Driver should not fall asleep at the wheel	Frontal view: detect yawning	False + <15% False - <5% Error rate <20%	1% 0% = 1%	
Driver should not fall asleep at the wheel	Frontal view: detect eyes closed	False + <15% False - <25% Error rate <40%	19% 27% = 47%	
Driver should never take eyes off the road for > 2 secs	Frontal view: Eyes looking away > 2 sec	False - <40% Error rate <40%	80% = 80%	
Driver should not fall asleep at the wheel	Side view: turned > 2 secs	False + <30% False - <5% Error rate <35%	Stretch goal	
Driver should never take eyes off the road for > 2 secs so computation must be fast	Latency < 1000ms	Measure the time between action and output	183.3 ms	
Driver should never take eyes off the road for > 2 secs so computation must be fast	>= 5 frame/sec (fps)	Run the program with a person looking at the camera for 60 seconds. Measure the frame rate	5.70 fps	
Total Device Accuracy	-	Accuracy of model against test suite >= 75%	85.9%	