Vehicle Accident Prevention System (VAPS)

Capstone - Final Presentation

Team

Dr. Khalid Hafeez

Jasindan Rasalingam

Kaushal Patel

Darron Singh

Ashwin Kamalakannan

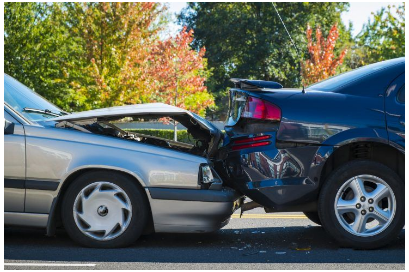
Introduction

Problem Statement: Drivers need a cheaper, alternative tool to help prevent accidents on the road
Project Objective: Develop a system to prevent avoidable vehicle collision scenarios

2

Major Components

Distracted Driver Detection



Proactive Collision Detection

Existing Solutions

Modern Car

- Collision Avoidance
- Expensive

Dash Cam

- Cheap
- No Features

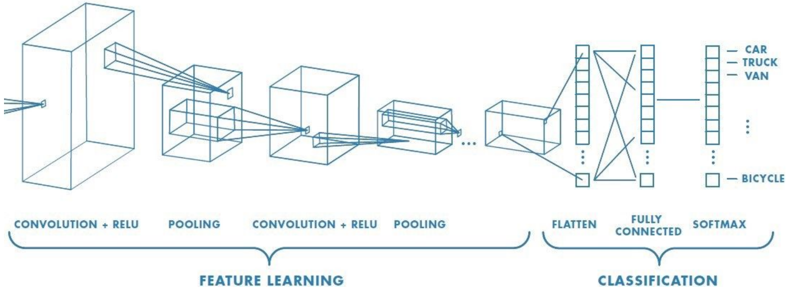
Soteria

- Provides DDD
- No definitive timeline for the future
- Provides proof of solution

Distracted Driver Detection

Component Objective:
Detect distracted behaviour portrayed by the driver, alert when necessary

Talking Right



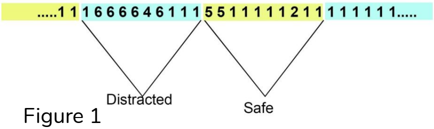


Figure 1

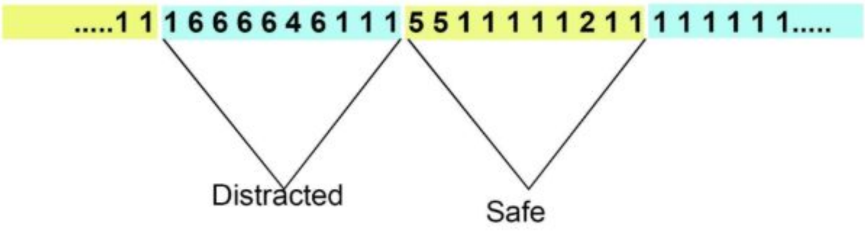
If consistent distracted state prediction

Source Code: https://github.com/Artemis854/Distracted-Driver-Detection
Dataset: https://www.floydhub.com/fastai/datasets/kaggle-state-farm-distracted-driver-detection

Internal Module Design Process

Module	Requirement
Internal	Detected distracted and not distracted state accurately (80%)
	Produce ALERT trigger if driver is distracted for more than 6 out of 10 frames from video feed

- Concluded on using convolutional neural networks
- Used datasets from Kaggle competition, skipped data collection

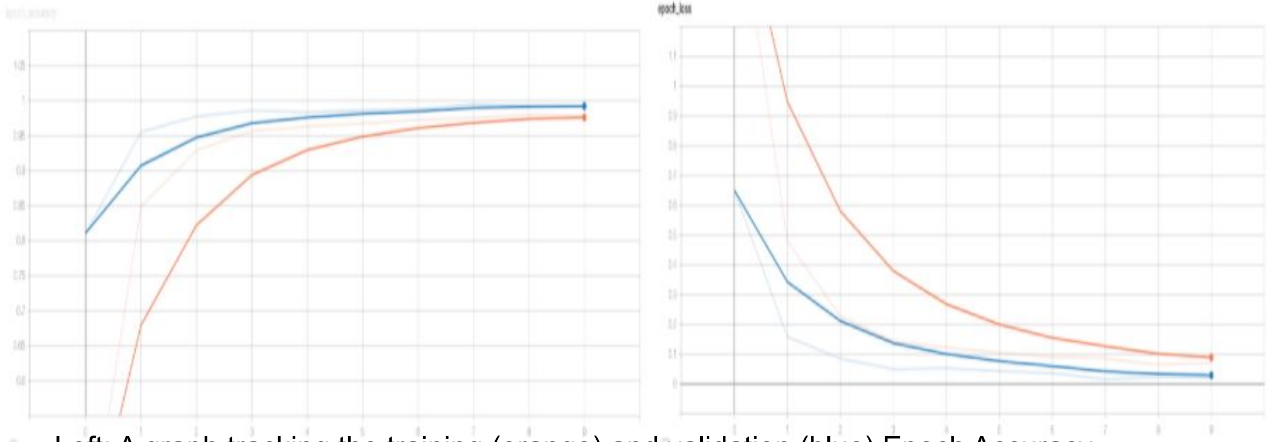


Abstraction of distracted driver detection algorithm

Internal Module Testing and Results

CNN Model Metrics

- Training results for the deep learning model had an accuracy of 90%
- Validation set was higher than training set



Left: A graph tracking the training (orange) and validation (blue) Epoch Accuracy

Right: A graph tracking the training (orange) and validation (blue) Epoch Loss

Internal Module Acceptance Testing

- Consisted of pre-recorded clips of various distracted behaviors

- Used confusion matrix
  - Can analyze the apparent accuracy of detection model
- Provided with the recall rate and precision for positive accuracy
- Specificity rate and negative predictive value for negative accuracy
- Wanted to ensure that an average detection was below 0.2s

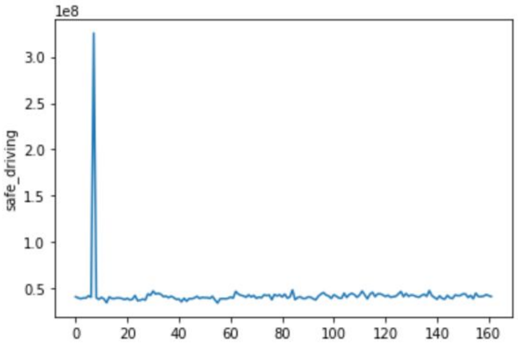
Example of a processed frame from the detection algorithm

Mappings for Confusion Matrix from Processed Clip	
True positive Driver is distracted and Prediction is distracted	**False positive** Driver is distracted and Prediction is not distracted
False negative Driver is not distracted and Prediction is distracted	**True negative** Driver is not distracted and Prediction is not distracted

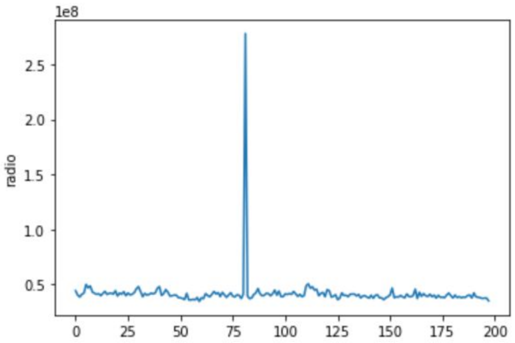
Internal Module Performance Testing Continued

- Average Precision: 85%
- Recall Rate: 81.75%
- Certain individual cases were below the benchmark of 75%

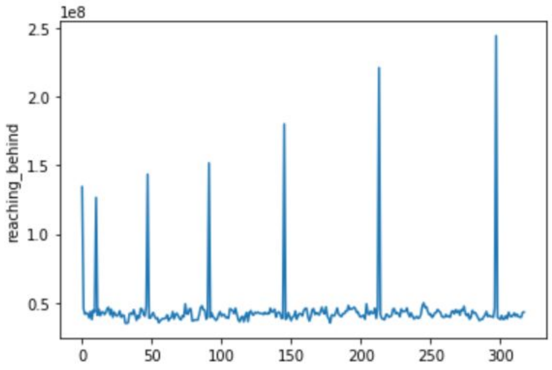
Processing Results	Time (ms)
Average Time	44.24ms
Min Time	34.89ms
Max Time	244.57ms



Safe Driving



Radio



Reaching Behind

Vehicle Detection

Objective
Detect vehicles approaching and alert driver if necessary

python

OpenCV

STEREOPI

Pedestrian Detection

Objective
Detect pedestrians in front of driver and alert when necessary

Depth Map

https://stereopi.com/

External Module Acceptance Testing and Results

Scenario	Accuracy	Average Time
Slow - Car Rear	>95%	0.76s
Medium - Car Rear	>90%	0.78s
Fast - Car Rear	>80%	0.74s
Pedestrians	>95%	0.65s

- Average detection rate of 0.2 seconds when lowering resolution of video input

Future Work

- Due to technological limitations, unable to combine the two modules into one whole system
- Object detection distance can be increased to a more suitable distance in order to warn drivers
- False positives in both external and internal modules need to be ruled out
- Smaller and more easy to install hardware
- Higher performance hardware or more optimized software

Demo Time

https://www.youtube.com/watch?v=hlfcGbhFu8g