BLINKAH Final Test Plan (Team 02)

Parker Van Roy (parkervr@bu.edu)
Raghurama Bukkarayasamudram (rbukka@bu.edu)
Ritam Das (ritamdas@bu.edu)
William Pine (wpine@bu.edu)
Ayush Upneja (upneja@bu.edu)

April 2021

Contents

1 Required Materials **3**
- 1.1 Hardware . . . 3
- 1.2 Head Unit Software . . . 3
- 1.3 BCF . . . 3
- 1.4 Other . . . 3

2 Introduction to Project **4**

3 Test Plan **5**

4 Set-up **6**
- 4.1 Server-side . . . 6
- 4.2 Head-unit side: . . . 6

5 Testing Procedures **7**

6 Scoring Criteria **8**
- 6.1 Head Unit: . . . 8
- 6.2 BCF . . . 8

1 Required Materials

1.1 Hardware

- Jetson Nano 4G
- CSI Camera
- 7" HDMI LCD Touchscreen
- USB Speaker
- Hologram 4G Modem & SIM

1.2 Head Unit Software

- OpenCV ALPR
- OpenCV MobileNet V2
- OpenCV Proprietary Lane Detection
- Erratic Driver Detection
- QtPy5 User Interface
- IBM Watson TTS Audio Notifications

1.3 BCF

- Nodejs
- Django
- Firebase
- 2GB Shared Intel SSD Digital Ocean Droplet

1.4 Other

- Car: 1998 Volvo V70 Cross-Country Station-wagon (AKA BLINKAH Mobile Transporter)

2 Introduction to Project

What is “BLINKAH?” BLINKAH Head Units are edge-computing dash cameras that leverage computer vision to analyze and report on other vehicles on the roads.Connected to the BLINKAH Computational Frontend, we analyze data from multiple head units and keep track of vehicle behavior, allowing us to create reports of erratic drivers for insurance purposes and to send warnings to drivers in the vicinity. BLINKAH ultimately eliminates irrelevant entities in traffic stops and reinforces safe driving habits.

3 Test Plan

In this project, we plan to breakdown the test into two portions: the BLINKAH Head Unit and the BLINKAH Computational Frontend. The BLINKAH head unit involves the jetson nano, its required hardware components, interaction with the cloud, and is expected to perform the following:

- Stream road footage
  - Confirm stream functions locally
- Computer vision subcomponents
  - ALPR test to accurately read license plates from video
  - MobileNet V2 to correctly identify other vehicles
  - Confirm accurate lane detection
- Detect Erratic Driving
  - Set and track vehicle bounds
  - Simulate Vehicle position
  - Detect report swerving
- Head Unit Frontend
  - Test vocal notifications
  - Show CV Output
  - Ensure working 4G Communication

In addition to these elements, the BLINKAH head unit is also responsible for sending the stream footage and incident report to the backend.

The BLINKAH Computational Frontend is a dashboard that polls backend data for informative visualization purposes. It is expected to perform the following:

- Poll Chronological Backend Data
  - Reported in tabular format
  - Visualize incident locations on heatmap
  - License Plate Lookup
- Administrative Dashboard

4 Set-up

4.1 Server-side

We will be hosting the Django server that needs to enable ssh connectivity in order for the head unit to communicate with the backend.

- Create a virtual environment
- Run python manage.py runserver @ip_address:PORT

4.2 Head-unit side:

We will be mounting the jetson nano and its components to the dash of the BLINKAH mobile, making sure that the camera has an optimal viewing angle for roads and vehicles. Power the Jetson and run the UI application.

5 Testing Procedures

To properly test our project, we plan to drive around select neighborhoods in Boston to gain multiple road perspectives, road conditions, and a variety of vehicles. We will test our computer vision and erratic driver detection algorithm on these vehicles and roads to generate real time reports. We expect to see all reports listed in the Django reports tab, and while all of them may not qualify as erratic driving, we expect at least of our reports that can be classified as erratic driving incidents. These reports are polled and received by the BLINKAH Computational Frontend where it should display the report along with the incident locations visualized on a heatmap.

6 Scoring Criteria

A successful final test should pass the following requirements and succeed in basic operation.

6.1 Head Unit:

- Properly stream road footage send to server
- Perform the following CV components properly
  - Collect license plates from ALPR
  - Detect vehicles from MobileNet V2
  - Detect lanes from OpenCV
- Determine if vehicle qualifies as an erratic driver
  - Report incident to BLINKAH Cloud
  - Confidence value consistently $> \frac{2}{3}$
- Frontend
  - Ensure functional UI
  - Audio notifications of incident(s) functional

6.2 BCF

- Server is working and polling incident data properly
  - Display chronological incident report in tabular format
  - Visualize incident locations on heatmap
  - License plate lookup
- Functional administrative dashboard

Task	Category	Comments	Result %
Stream Road Footage & Send to Server	Head Unit		
Collect License Plates (ALPR)	Head Unit		
Vehicle Detection (MobileNet V2)	Head Unit		
Lane Detection (OpenCV)	Head Unit		
Erratic Driver Algorithm	Head Unit		
Report Erratic Driver Incident to Backend	Head Unit		
Ensure Consistent Confidence Value > ⅔	Head Unit		
Functional Head Unit UI	Head Unit		
Audio notifications	Head Unit		
Server running properly	BCF		
Server polling incident data properly	BCF		
Display chronological incident report(s) from database	BCF		
Incident Heatmap Visualization	BCF		
License Plate Lookup	BCF		
Functional Admin Dash	BCF		
Result %	→	→	