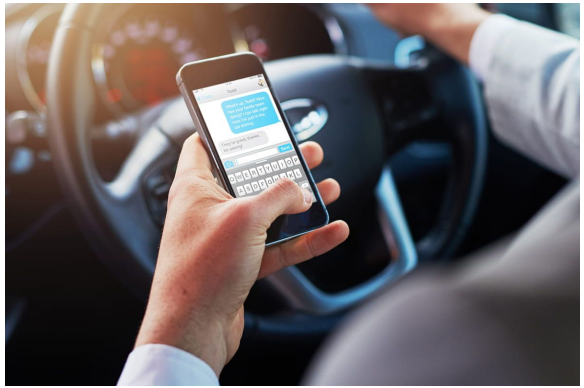
Detecting Texting while Driving

Kristina Robinson
Andrew Knaack
Sara Mace

Lucas Golinghorst
Derek Clayton
Ryan Baker

1. Problem Statement

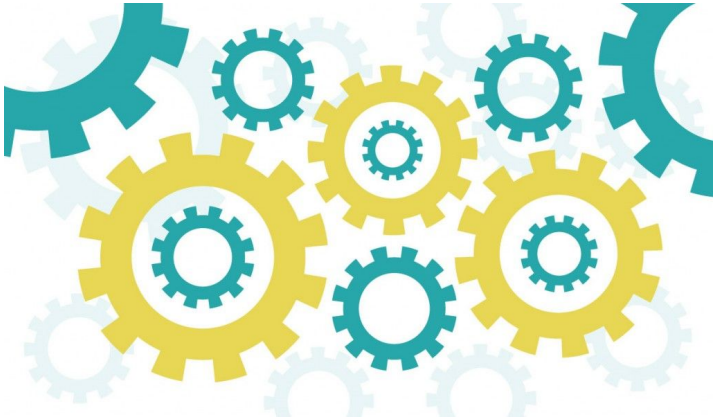
- Addresses the issue of texting while driving.
- 3 years ago, 391,000 injuries were caused by distracted driving accidents.
- Our solution is to build an android application to detect if someone is texting while driving.
- Our goals include detecting whether someone is in the driver's seat.
- In order to do that without simply locking out everyone from their phones, our solution will have to include many different measures to ensure accurate detection.



2. Functional / Non-Functional Requirements

Functional

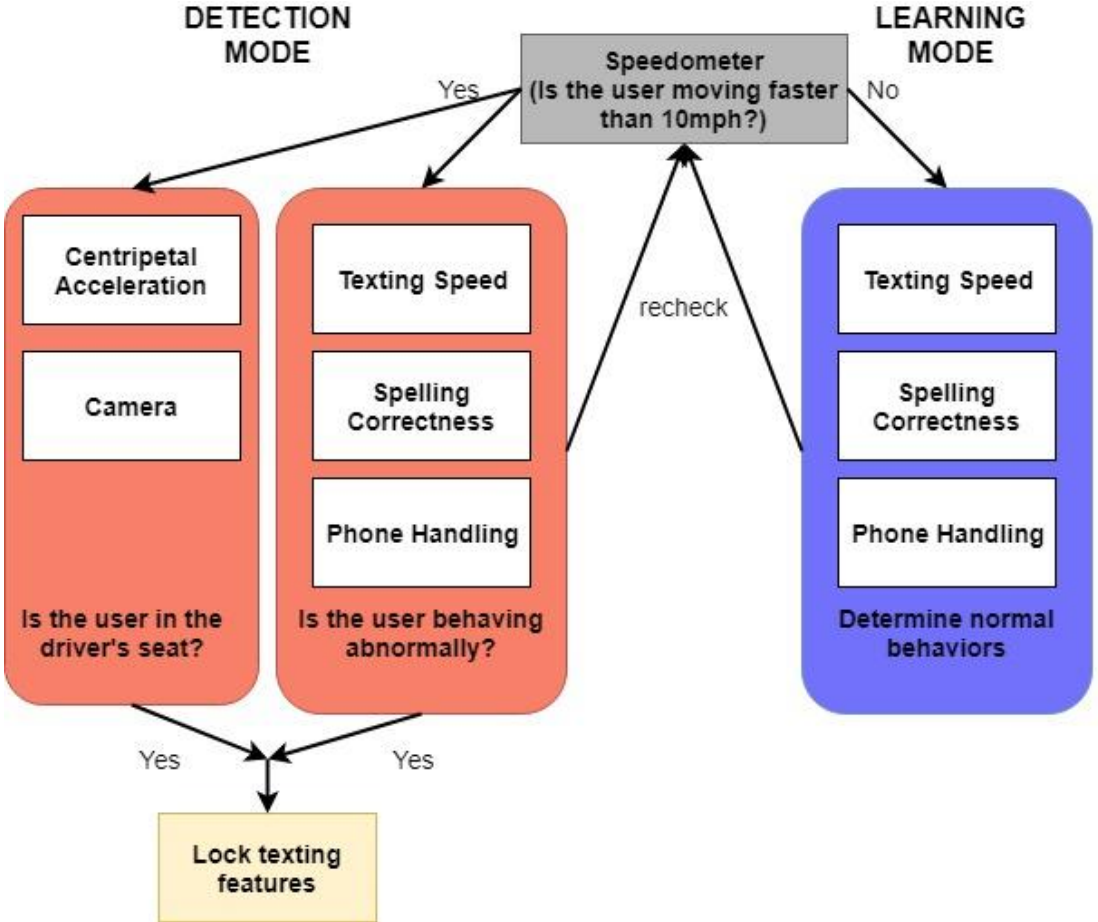
- Prevent users from texting if they are driving
- Error rate <= 10% , including false positive/negatives
- Android OS 6.0 and newer.
- Does not require Internet



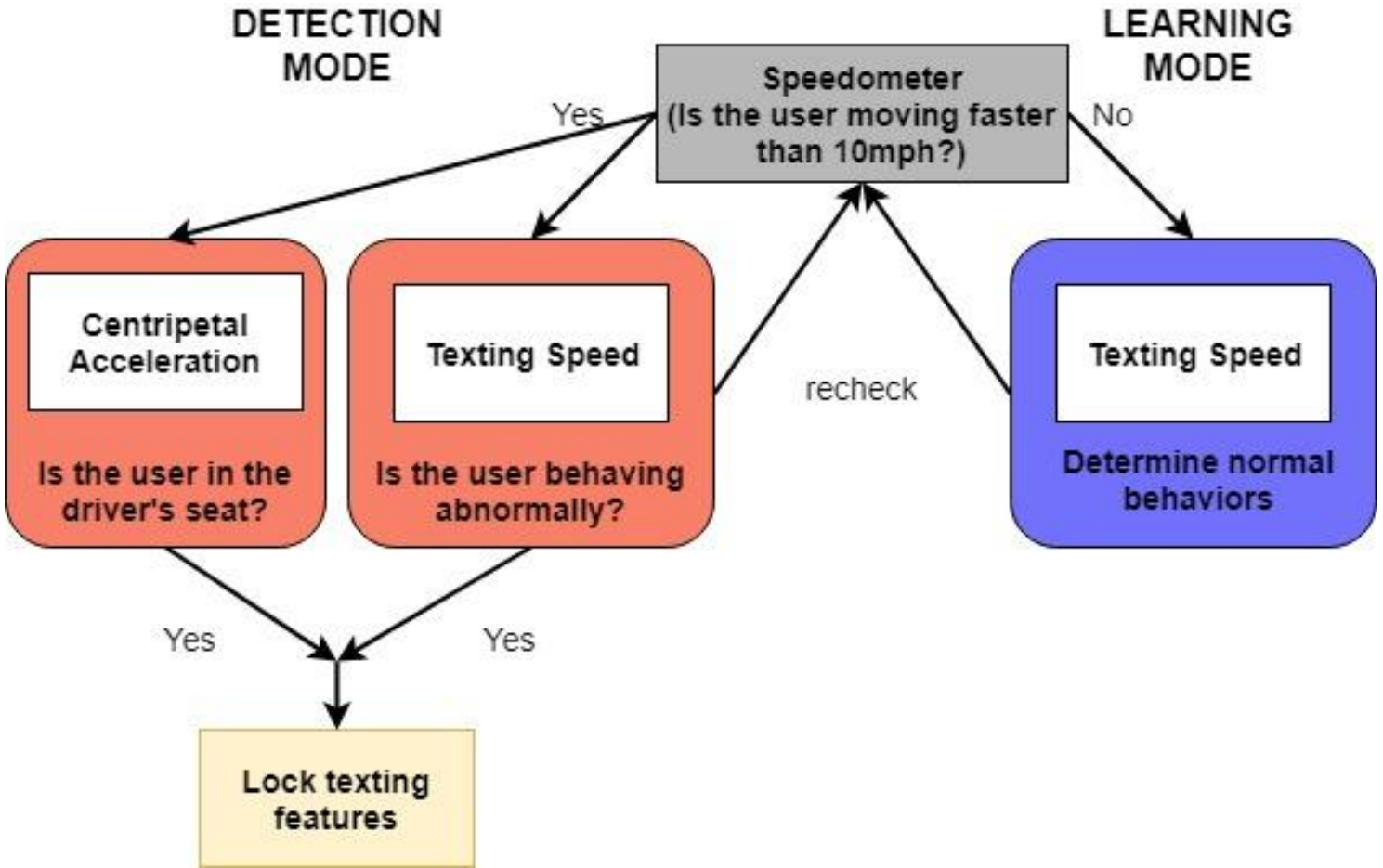
Non-Functional

- Easy to use
- Does not crash at least 95% of the time
- Comply with applicable sensitive data laws

3. System Design (planned)



3. System Design (implemented)

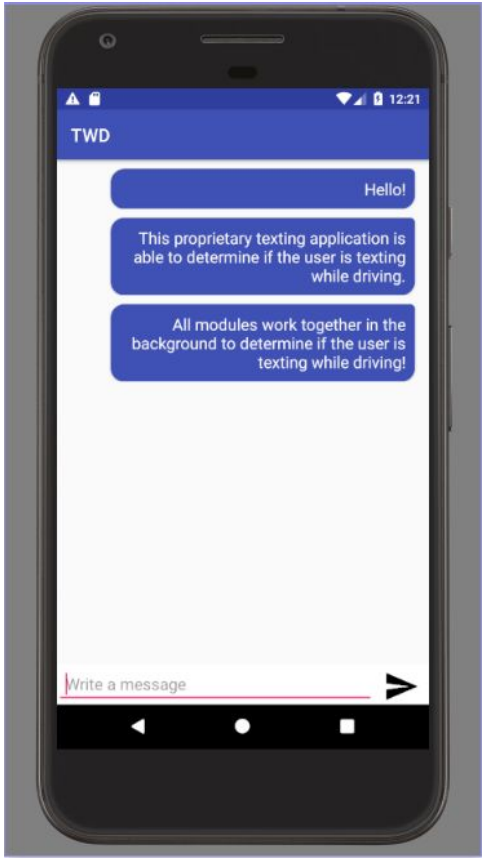


4. System Implementation

I. Proprietary Texting Application
II. Speedometer
III. Texting Speed
IV. Centripetal Acceleration
V. Spelling Correctness (Failed)
VI. Phone Handling (Failed)
VII. Camera Recognition (Failed)

Proprietary Texting Application

- Was built because we couldn’t use the built in text messaging application
- Application allows user to type and send messages which works with modules to determine texting while driving

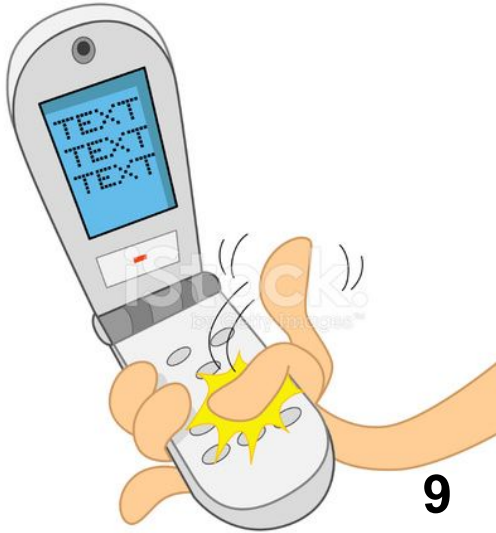


Speedometer

- Location Services API
  - Uses GPS and (if possible) Wi-Fi signals
  - Updates every ~10 seconds
- Provides getSpeed() method for main activity
- Target danger speed is above 10 mph

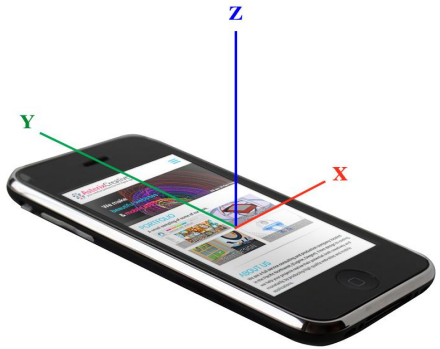
Texting Speed

- Text Watcher API
- Learning Mode: continuously update user’s average texting speed
- Detection Mode: compare to known average



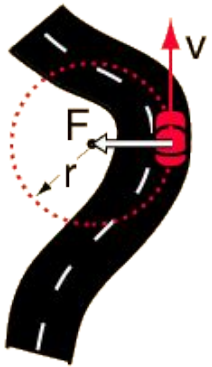
Centripetal Acceleration

- Driver’s and passenger's experience different levels of centripetal acceleration during turns.
- Using the gyroscope and magnetometer, it is possible to determine if the user is on the driver’s side or the passenger’s side.
- Module detects turns using the gyroscope.
- Turn amplitude calculated using magnetometer readings.



$$F_{centripetal} = m\frac{v^2}{r}$$

$\frac{v^2}{r}$ is the centripetal acceleration

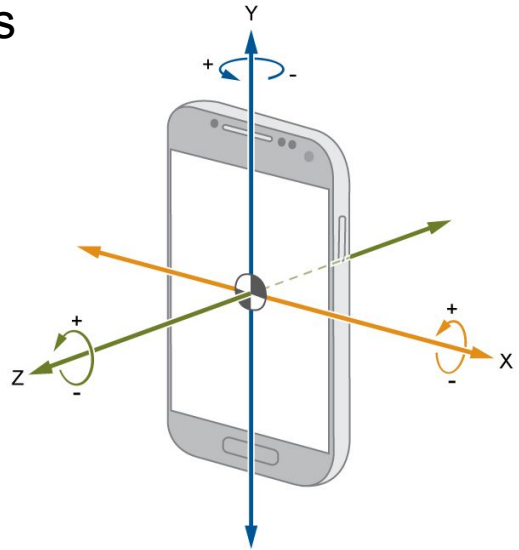


Spelling Correctness

- Has built-in Android API Spell Checker Session
- Unfortunately, has been disabled on some phones for a few years now
- NOT VIABLE

Phone Handling

- Gyroscope and Magnetometer were evaluated to look at differences in how the user holds their phone
- Both sensors did not give a large enough difference to determine differences in normal phone behavior and texting behaviors
- Not a viable module



Camera Recognition

Open-source libraries for training machine learning algorithms dedicated to image recognition

Tensorflow- Did not work because...

- Does not allow training of only one class
- Requires several old programs

OpenCv - More precise processing power (using Bitmapping), but...

- Less developed API tools, no built-in image classes.

Overall, required more data images of seatbelts (~10,000) to create a defined target image, and a software library that had training tools to build the seatbelt class.

5. Module Testing & Evaluation

Speedometer

- Printed out recorded speed and drove around with another person to compare recorded speed to actual speed
- Test zones: Ames, Des Moines, I-35

Texting Speed

- Played Mario Kart to simulate distracted driving
- Typed predetermined messages to see if the user would be locked out after dropping below 70% of normal texting speed

Centripetal Acceleration

- Made consecutive left turns or right turns in a parking lot
- Recorded number of times driver and passenger were correctly identified
- Turns are determined correctly 100% of the time
- Identification of user location is correct 54% of the time

5. Integration Testing & Evaluation

- Each module was independently tested to make sure that they would block a user from texting if certain thresholds were met
- Speedometer: if the speed reached 10 miles per hour or above the user would be blocked from texting
- Texting Speed: if texting speed dropped below 70% of their average speed, the user would be blocked from texting
- Centripetal Acceleration: If the user was identified as a driver, they were blocked from texting

5. System Testing & Evaluation

- All modules were considered for blocking a user from texting if they met the appropriate thresholds
- Tested by driving around a parking lot making left and right turns and imitated distracted behavior.
- Reached our fourth milestone of false positive and false negative rates at or below 50%
- False positive rate: 0%
- False negative rate: 40%

6. Use-Case Scenarios

Speed > 10mph	User is Driver	Distracted Texting	Texting Lock
F	F	F	No
F	F	T	No
F	T	F	No
F	T	T	No
T	F	F	No
T	F	T	No
T	T	F	No
T	T	T	Yes

7. Conclusions

- It is possible to determine if someone is texting while driving.
- Future improvements:
  - Machine Learning
    - Use Texting Speed and Centripetal Acceleration
    - Take into account different phones and different users
  - Determine if the user is sitting in the back seat or the front seat
    - Currently only determines if in driver or passenger side of car