Moving Traffic Obstacle Detection using Smartphones

Xiangzhu Long, Tierra Bills, Aisha Walcott, John Wamburu, Reginald Bryant
IBM Research Africa
Catholic University Campus Langata Rd, Nairobi, Kenya, 00200

Abstract

In cities in developing countries like Nairobi, Kenya, it is common for roadways to be shared by a variety of transport modes because of limited pedestrian infrastructure. These modes include personal vehicles, trucks, two-wheelers, bicycles, pushcarts, cattles and pedestrians. This creates situations where unexpected slow moving traffic forms what we call "Moving Traffic Obstacle" (MTO) causing faster moving vehicles to frequently maneuver in order to overtake slower obstacles. Given the significant impact of MTOs on traffic conditions and incidents, we propose a method of detecting the presence of MTOs by measuring the sequences of driver maneuvers from several vehicles following one another with smartphone. The methodology is as follows: (1) build decision tree model to classify vehicle maneuvers by selecting features calculated on accelerometer, gyroscope, speed and time parameters, (2) predict the potential presence of MTOs due to abnormal vehicle maneuvers, and (3) build MTO detection by clustering these potential moving traffic obstacles. While there are many tools for observing static obstacles (i.e. potholes, manholes, etc.), this study takes the first step towards a tool for detecting MTOs.

keywords: Traffic, smartphone, developing country, decision tree, k-means

I. Introduction

Traffic congestion in developing cities like Nairobi, Kenya can be significantly impacted by the presence of "Moving Traffic Obstacles" (MTOs) [1]. These MTOs are those events that temporarily exist on the road, moving with or against the direction of traffic at slower speeds. As shown in Figure 1, they include two-wheelers, pushcarts, animals, and pedestrians, which have quite different influence on traffic compared with static obstacles, such as potholes and speed bumps.

Figure 1: MTO Examples

The existing tools [2] that can detect road obstacles can be prohibitively expensive and sophisticated, which are unpractical for developing countries. Smartphones are wide spread in Kenya and extremely valuable in this case because they are relatively low-cost, ubiquitous, and carry a suite of sensors that can be used to measure the movement of vehicles. As seen by recent innovations [3, 4, 5, 6, 7], smartphones have great potential to improve traffic quality and safety.

This paper is organized as follows. Section II introduced the overall framework of the system. Section III describes the related studies on the field of detection tool and traffic detection with smartphone sensors. Section IV defines the maneuver, describes data collection and data labelling, and presents methods for building maneuver classifier. Section V details the transfer from MTO matrix to potential MTO (PMTO) matrix, and describes MTO detection by clustering PMTOs. Section VI discusses the results based on the experiments. Conclusions are given in section VII.

II. Overview

As the overall framework shown in Figure 2, we first collected labeled data with smartphone sensors for vehicle maneuvers to training a decision tree-based maneuver classifier. This model is implemented to classify maneuvers for the sensor data from several vehicles following one another. By measuring the abnormal maneuvers, we predict PMTOs and build PMTO matrix. Based on k-means clustering on these PMTOs, we detect the presence of MTOs.

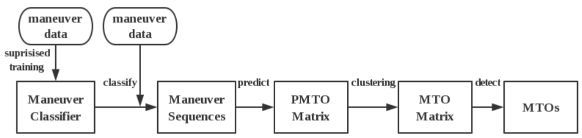

Figure 2: Overview Framework

III. Related Work

Road quality measurement systems can play a significant role in improving traffic conditions and incident rates. Many studies have focused on road quality, often requiring dedicated vehicles equipped with expensive and sophisticated hardware [2]. This may be unrealistic to implement in developing countries. Smartphone technologies provide a solution to this challenge, and have been applied across traffic domains, such as traffic safety [3] and road management [4].

There are several studies on the analysis of the traffic quality, driver behaviors, and hazard detection based on different sensor data as parameters from smartphones. V. P. Tonde et al. described a road irregularity detection based on smart phone sensors [4]. GreenRoad is a mobile phone-based product that can provide real time advice on driver behavior. J. E. Meseguer et al. proposed a DrivingStyles platform that combined neural networks to access driver behaviors with smartphones [5]. C. Thompson et al. presented an accident detection system that can provide emergency awareness and help to reduce the traffic congestions [6].

Our solution differs from preceding research by providing detection of MTOs, which is often ignored in developed countries, yet helps to improve the traffic quality in developing countries.

IV. Maneuver Classifier

As shown in Figure 3, there are three events happened in the traffic: no obstacles on the road, MTOs, and static obstacles. With no obstacles, drivers usually have the same driver behaivors, such as driving at the same speed, or making a turn. The probability that some drivers take erratic behaviors is so small that can be excluded by the majority. When there is a MTO, the driver will take deviant action to avoid it. For example, to bypass a pedetrain, some drivers will brake, some may make a turn, and others just drive slowly behind the pedetrain. For the third situation, to avoid a static obstacle, driver behaivors may varied from the types of the static obstacle and different drivers. For example, if there is speed bump that crosses the street, usually drivers will brake and then drive through it; if there is a stone of middle size, different drivers may take different behaviors. Thus, the algorithm in is paper relates the MTOs with differnt types of driver maneuvers.

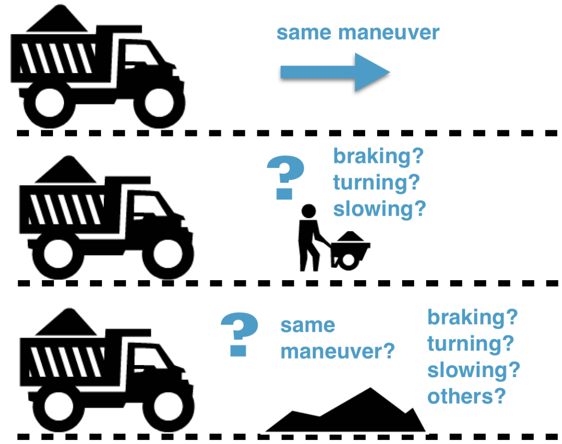

Figure 3: Traffic Events

Maneuver Definition

From the perspective of driver's reaction to MTOs, we define four different vehicle maneuvers that can represent the driver reactions to MTOs [7], two of which are shown in Figure 4, the third one is "slowing", and the fourth is "others".

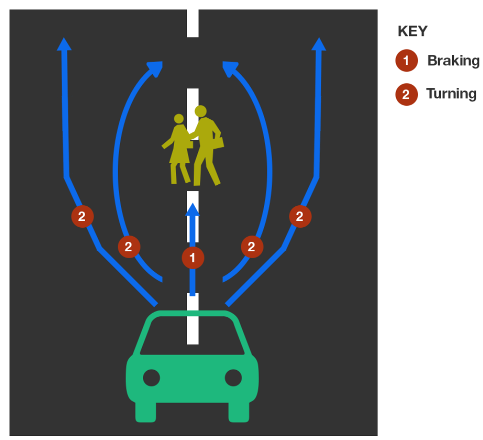

Figure 4: Maneuver definition

"Braking" means decreasing the speed to stop in 1 second. "Turning" comprises all kinds of turns, includes "left turn" - turning left with about 45° and driving straight, "right turn" - turning right with about 45° and driving straight, "left swerve" - turning left, then turning right without changing the lane and "Right swerve" - turning left, then turning right without changing the lane. "Slowing" means that the vehicle drives so slowly that forms moving bottleneck [8]. And "others" comprises other unlabeled events.

Data Collection

IBM StreetSense is a mobile application built on the Android platform that is designed to collect sensor data on road surface conditions. The mobile device is fitted onto a vehicle and as it moves, the application records data on different sensor readings. We collect three-axis data from the accelerometer, gyroscope, gravity, orientation, linear acceleration and magnetic field sensors at a frequency of 20 Hz.

As the process of data collection shown in Figure 5, sensor data is then mapped onto specific locations using location information obtained from GPS, and uploaded to a central NoSQL database for storage. A backend application pulls this data for processing. We take the data through a four-step pre-processing series: data cleansing, reorientation, map matching and segmentation. The processed data is then persisted into a relational database extended with geo capabilities for further analysis.

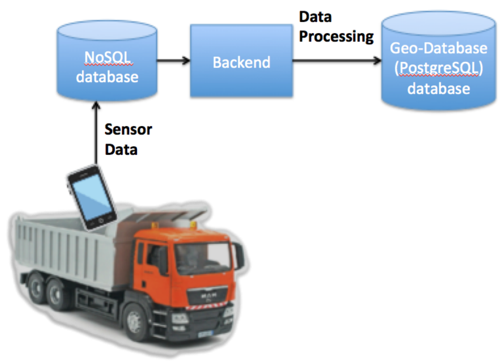

Figure 5: The Process of Data Collection

In this experiment, we mounted smartphones equipped with IBM StreetSense on 9 garbage trucks (as shown in Figure 6) from Kaloheni Depot in Nairobi, Kenya to collect sensor data during their daily working.

Figure 6: Garbage Truck

Data Labelling

As shown in Figure 7. We collected labeled data as we sat in a separate vehicle that was following a garbage truck. The garbage truck was equipped with a smartphone to receive sensor data. We used a mobile app, which can record time stamps, to label the maneuvers. When garbage truck make a maneuver, we labeled it at the start and end of each maneuver. We extract time stamps of starting and ending at labels for one maneuver.

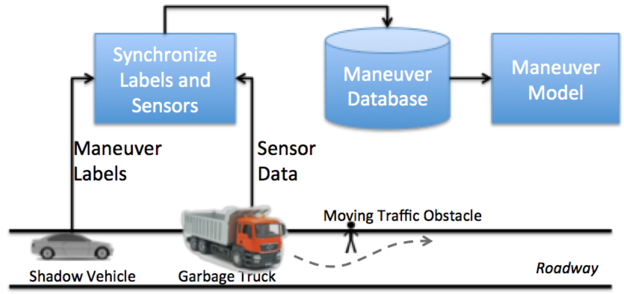

Figure 7: Data Labeling

We synchronized labels by matching starting and ending time with sensor data and extracting these data as one observation. The algorithm of labelling data is shown as Algorithm 1.

Algorithm 1: Data Labelling.

Input: labels file and sensor data
Output: a collection of several files that stores observations of each category

# Get [starting, ending] time stamps
for $i_1 \leftarrow maneuver\_names$ **do**
  $TS[i_1] \leftarrow getStartEndTS(labels\_file, i_1)$
# Synchronize the labels with sensor data
for $i_3 \leftarrow 1$ *to* 6 **do**
  for $[ts1, ts2] \leftarrow TS[i_3]$ **do**
    csv ← getData(sensor_data, ts1, ts2)
    # Put csv to the file under its category
    file ← label_name
    file ← csv
return *all_files*

Classifying Maneuvers

Based on the labeled data as training data, we select three sensor data – accelerometer, gyroscope, speed – and the time duration of a maneuver as our attributes, because they are directly casued by manuvers, and themselves have less ovelapping relationships. For each observation, we calculate the maximum, minimum, mean, standard deviation of each sensor parameter and combine time duration as features. We build the basic classifier by decision tree with an optimal accuracy of 84.6% on 9-fold cross validation, and its confusion matrix shown as Figure **??**, which mentions that the classifer has low recall as it does not have high error rate of misclassifying certain types of maneuvers. Key attributes in decision tree rules are standard deviation of z-axis gyroscope, mean of y-axis gyroscope, maximun of speed, standard deviation of z-axis accelerometer, time duration. The algorithm of building maneuver classifier is shown as Algorithm 2:

True Diagnosis / **Screening Test**

	①	②	③	④
①′	8	0	1	2
②′	0	3	0	1
③′	0	0	8	0
④′	3	0	0	19

Figure 8: Confusion Matrix

Algorithm 2: Maneuver Classifier.

```
Input: a collection of several files that stores
       observations of each category
Output: maneuver classifier
for i ← 1 to len(all_files) do
    csvs ← getAllcsvs(all_files[i])
    label ← file_name
    for j ← 1 to len(all_files) do
        csv ← csvs[j]
        # Select accelerometer, gyroscope, speed, and
        time stamps of a maneuver as attributes
        csv_selected ← getAttributes(csv)
        # Calculate the maximum, minimum, mean,
        standard deviation of each sensor data and
        combine time duration as features
        csv_caculated ← getFeatures(csv_selected)
        # Get the label
        csv_caculated[label] ← label
        training[i] ← csv_caculated
manuever_classifier ← buildDecisionTree(training)
return manuever_classifier
```

V. Moving Traffic Obstacle Detection

This section shows the method of data pocessing and details the algorithm of building MTO detection based on PMTO predictor as shown in Algorithm 3.

Data Pocessing

We select sensor data collected from 9 vehicles along Jogoo Road from Kaloheni Depot to the nearest traffic circle in Nairobi, Kenya on the 10th of May 2015 and Landihies Road on the 7th of July 2015. These vehicles followed one another about 4 meters. Jogoo Road is where Kaloheni Depot is located, and because garbage trucks set off from this road, it has great potential to have data where vehicles are following each other. There are many MTOs on Landihies Road, as it is densely populated, and contains different modes of transportation.

Sample data are windowed according to a distance of 4 meters, as this is the average distance required for a car maneuver and we get it from the data for building maneuver classifier. In addition, the windows are overlapped by approximatly 3 meters to reduce missing maneuvers. When

Algorithm 3: Moving Traffic Obstacle Detection.

```
Input: a collection of csv files that stores sensor data,
       and maneuver classifier
Output: MTOs matrix
# Process the sensor data
testing ← processData(csv_files)
n_vehicles ← len(csv_files)
# Predict maneuvers
for i1 ← 1 to n_vehicles do
    manuevers[i1] ← predictManuevers(testing[i1],
    classifier)
# Get PMTO matrix
n_windows ← len(manuevers[0])
for i2 ← 1 to n_windows do
    PMTO_matrix[i2] ← getPMTOs(manuevers[:][i2])
# Clustering PMTOs
PMTO_points ← getPoints(PMTO_matrix)
n_clusterings ← sum(PMTO_points) / n_vehicles
clusterings ← KMeans(PMTO_points, n_clusterings)
# Dectect MTOs
MTO_matrix ← getMTOs(clusterings)
return MTO_matrix
```

there is low GPS accuracy, we interpolate by assuming a uniform distance between the points where GPS failed then recovered. The feature processing for sample and training data are the same.

Potential Moving Traffic Obstacle

We can exclude the maneuvers that taken under the situations of no obstacles by first predicting the maneuvers that potentially caused by MTOs. The method for predicting PMTOs is as following: (1) create a matrix of maneuvers from several vehicles following each other, (2) predict the maneuvers that potentially caused by MTOs by comparing maneuvers in matrix, and (3) create a matrix of PMTOs.

For 9 sequences of maneuvers from 9 vehicles following one another, we build the maneuvers matrix (example is shown in section 10(a)) for each road. We focus on the maneuvers that occur in the same location and predict each one as a PMTO if its percentage on the total number of maneuvers at this location is below a threshold, which can distinguish PMTOs. We difine this threshold as TPMTO.

For the static obstacles, different drivers might take different maneuvers. For example, when encountering speed bump, some might make a turn to avoid it, some might braking and drive over it, some may slow down and take it, and others may just drive through it. Thus, we are supposed to consider this situation that all the windows at the same location can be categoried to PMTOs and then exclude afterwards. To achieve it, we choose the maxmium percentage that can consider all the manuvers as PMTOs. For this experienment, the maneuvers sequence with average distribution is (1, 1, 1, 2, 2, 3, 3, 4, 4). We select the threshold as 34% by

$$ceiling(9/4) = 3,$$

$$ceiling(\frac{3}{9} \times 100) = 34(percentage).$$

For example, the sequence of maneuvers for these vehicles at Position 2 on Road shown at section 10(a) are (1, 1, 1, 1, 2, 0, 1, 1, 1), in which the percentage of 2 and 5 are both 11.1% (below 34%). Then we predict the maneuvers occurred for the fifth and sixth vehicles at Position 2 are potentially caused by MTOs. As an example shown in Figure 10, we build PMTOs matrix based on maneuvers matrix from these 9 vehicles.

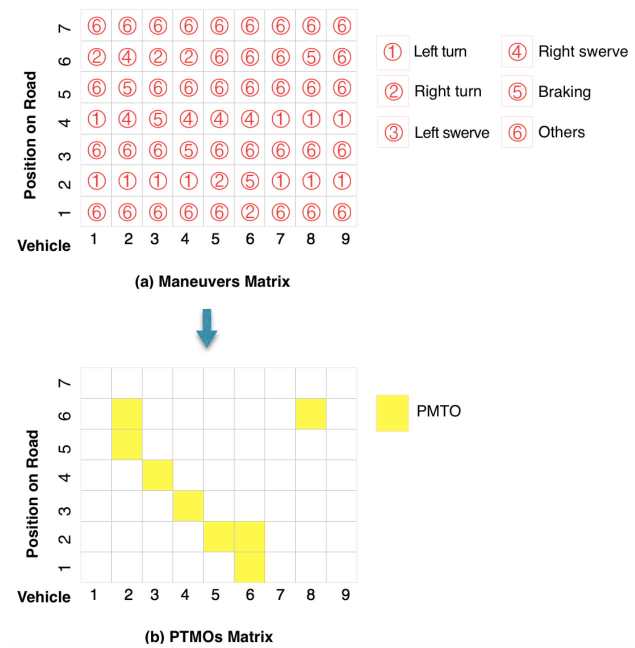

Figure 9: Maneuvers and PMTOs Matrixs

Detecting Moving Traffic Obstacles

As shown in Figure 10, the data at diagonal share the same time stamps, as the distance difference between vehicles is 4 meters, which is one window distance.

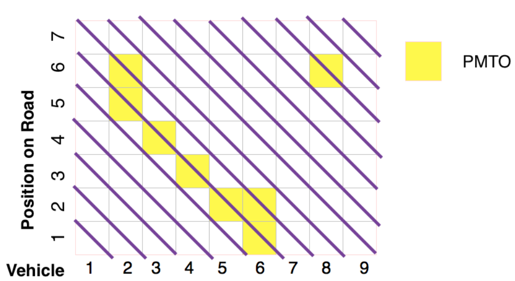

Figure 10: PMTOs Matrix Analysis

To exclude the PMTOs that caused by other types of static obstacles that may result in differnent types of maunuvers, the differences between the maneuvers caused by MTOs and static obstacles are described as following: referring to the situations shown in 11, (1) For MTOs, if the MTOs take the same or agaist the direction with the traffic, the maneuvers caused by PMTOs happen at different location. If the MTOs cross the street, for 9 vehicles in this experienment, not every vehicle will encounter it. (2) For static obstacles, the maneuvers caused by PMTOs happened at the same location. Thus, MTOs can be distinguished from PMTOs by the location where PMTOs happen.

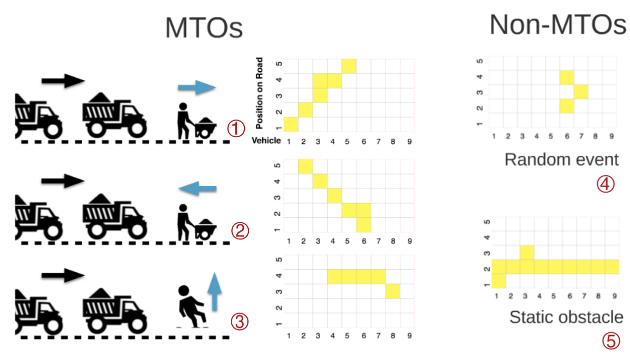

Figure 11: Possible Events Patterns

As shown in 11, there are three categories of events for MTOs. MTOs 1 represents MTOs that have the same direction with vehicles, because the continuous vehicles encounter PMTOs at forward location in increasing time stamps. MTOs 2 shows MTOs that against the direction, because the continuous vehicles encounter PMTOs at backward location in increasing time stamps. MTOs 3 is the one that across the road, because more than enough but not every vehicle encounter PMTOs at the same location in increasing time stamps. Non-MTOs 4 is detected as fake MTO because it contains few PMTOs that might caused by random events. Non-MTOs 5 does not belong to MTO because every vehicle encounter PMTOs at the same location, which means static obstacle exists here.

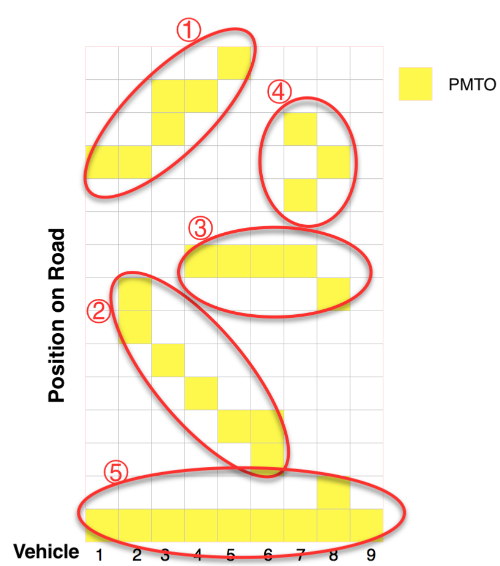

Figure 12: PMTOs Clustering

We make each PMTO as one point with vehicle number as its first dimension, and sequence number as the second dimension. For example, PMTO for vehicle 4 at its 3rd maneuver has a point position of (4, 3). Then we make k-means clustering on these points, with k is average number of PMTOs per vehicle,

$$k = \frac{\#(PMTOs)}{\#(Vehicles)},$$

$\#(Vehicles) = 9$ in this experiment.

For clusterings with not less than the average number of maneuvers per vehicle and in which not include every vehicle at the same location, we detect it as a MTO. For example, as shown in Figure 12, clustering 1, 2 and 3 are detected as MTOs while we ignore clustering 4. Because clustering 1, 2 and 3 contains 6, 7, and 5 PMTOs respectively, which are more than 4 (the half number of 9 vehicles), clustering 4 has 3 PMTOs, which is less than 4, and clustering 5 contains every vehicle at the same location.

VI. Result

The distribution of MTOs we detected on the road is shown in Figure 13 (A is Landhies Road, B is Jogoo Road). There are more MTOs at Landhies Road than at Jogoo Road, and turning road has more MTOs. Figure 13A represents 400 meters of road and 23 MTOs were recorded. Figure 13B represents 500 meters of road and 16 MTOs were recorded. Landhies road proved to be the more busy street as it averages one MTO every 17 meters, while Jogoo road is slightly less obstacle prone, averaging one MTO every 31 meters. This averages out to one moving traffic obstacle every 24 meters on both roads. Observation windows for this research is four meters, which means that on average every six car lengths a MTO presents itself.

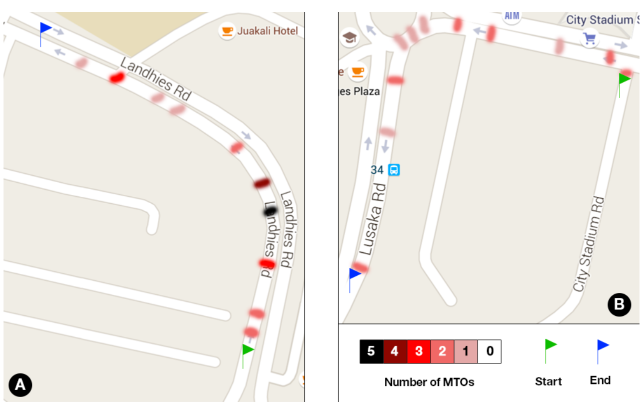

Figure 13: MTOs Distribution

VII. Conclusions and Future Work

This paper represents the first step to improve the traffic quality by detecting MTOs on the road. MTOs can significantly impact traffic conditions for developing countries but are largely ignored in the literature on traffic quality and traffic management. Our paper implemented a frugal and efficient method to detect these MTOs. First we build a classifier on six maneuvers by taking advantages of smartphone sensors, then we predict PMTOs by analyzing several sequences of maneuvers from vehicles followed one another on the test site, and the objective is to detect the MTOs by make clustering of these PMTOs.

Our next aims are to (1) equipped experimental smartphones on taxis, (2) collect a larger dataset of sensor data from smartphones on the pivotal roads in Nairobi, (3) analyze the distribution and influence of the moving traffic obstacles on the traffic, and finally (4) provide helpful tips for the government to improve traffic quality by managing moving traffic obstacles.

Acknowledgment

The authors would like to thank Michiaki Tatsubori and Skyler Speakman for their advice on this topic, and Andrew Kinai for his help on drawing Figure 4 and Figure 13.

References

[1] T. Bills, R. Bryant, and A. W. Bryant, "Towards a frugal framework for monitoring road quality," in *ITSC'14*, pp. 3022–3027, IEEE, 2014.

[2] K. Chen, M. Lu, G. Tan, and J. Wu, "Crsm: Crowdsourcing based road surface monitoring," in *HPCC_EUC*, pp. 2151–2158, IEEE, 2013.

[3] C.-W. You, M. Montes-de Oca, T. J. Bao, N. D. Lane, H. Lu, G. Cardone, L. Torresani, and A. T. Campbell, "Carsafe: a driver safety app that detects dangerous driving behavior using dual-cameras on smartphones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 671–672, ACM, 2012.

[4] V. Tonde, A. Jadhav, S. Shinde, A. Dhoka, and S. Bablade, "Road quality and ghats complexity analysis using android sensors,"

[5] J. E. Meseguer, C. T. Calafate, J. C. Cano, and P. Manzoni, "Drivingstyles: A smartphone application to assess driver behavior," in *ISCC*, pp. 000535–000540, IEEE, 2013.

[6] C. Thompson, J. White, B. Dougherty, A. Albright, and D. C. Schmidt, "Using smartphones to detect car accidents and provide situational awareness to emergency responders," in *Mobile Wireless Middleware, Operating Systems, and Applications*, pp. 29–42, Springer, 2010.

[7] D. Johnson, M. M. Trivedi, *et al.*, "Driving style recognition using a smartphone as a sensor platform," in *ITSC*, pp. 1609–1615, IEEE, 2011.

[8] J. C. Munoz and C. F. Daganzo, "Moving bottlenecks: a theory grounded on experimental observation," *University of California Transportation Center*, 2004.