Multimed Tools Appl (2016) 75:16959–16981
DOI 10.1007/s11042-015-2969-7

Determining driver phone use leveraging smartphone sensors

Yantao Li[1] · Gang Zhou[2] · Yue Li[2] · Du Shen[2]

Received: 21 March 2015 / Revised: 27 August 2015 / Accepted: 23 September 2015 /
Published online: 9 October 2015

Abstract Driver distraction by mobile phones has been a huge threat that leads to unnecessary accidents and human casualties, especially in hazardous road conditions. In this paper, we address a fundamental but critical issue of phone use during the driver behind the wheel. We propose, design and implement *SafeDrive* which achieves the goal of automatically determining driver phone use leveraging built-in smartphone sensors sensing driving conditions. We explore GPS and accelerometer sensors on smartphones to collect data, which can sufficiently capture driving conditions. With inputs of these data, we provide an accurate driving condition classification algorithm, that classifies driving conditions into five categories. Based on the classified driving conditions, *SafeDrive* makes a flexible control of driver phone use. We excessively evaluate the classification accuracy of our *SafeDrive* in local, highway, traffic jam, and complex conditions, respectively, and the results demonstrate that it can achieve up to 87 % classification accuracy in complex conditions.

Keywords Driver phone use · Smartphone · GPS · Accelerometer · Driving condition classification

1 Introduction

Nowadays, mobile phones and motor vehicles take significant parts in our daily life. Mobile phone users can be connected any time under any conditions, which greatly facilitates connections and communications and sufficiently reduces time cost, while motor vehi-

✉ Yantao Li
yantaoli@foxmail.com;liyantao@live.com;yantaoli@swu.edu.cn

[1] College of Computer and Information Sciences, Southwest University, Chongqing 400715, China

[2] Department of Computer Science, College of William and Mary, Williamsburg, VA 23187, USA

cles make it convenient for people to travel any place under any environment. Although people are enjoying the fast communication and convenient transportation brought by mobile phones and automobiles, they are also facing safety threats while driving, especially when a mobile phone is in use. Using mobile phone while driving (phone distraction) has been a major factor in crashes (automobile accidents) that has led to 415 fatalities (378 crashes, 12 % of fatalities in distraction-affected crashes) and 28,000 injuries (7 % of injured people in distraction-affected crashes), according to Distracted Driving 2012 from National Highway Traffic Safety Administration released in April 2014 [37]. Given that the high rate of mobile phone in use while driving contributes to vehicle accidents, many states have banned certain usage of phones in US, where 12 states prohibit drivers from using hand-held phones, and 43 states have banned text messaging when driving [10].

In fact, people are likely to use their mobile phones even while driving, such as emergency calls and incoming calls. For convenience, various hand-free devices, such as Buletooth and wireless earphones, are developed and implemented on motor vehicles, but these devices do not really help drivers reduce the risks of distraction-affected crashes [12, 21]. Therefore, how to assist drivers to make appropriate decisions of phone use (such as incoming/outgoing phone calls) while driving is a crucially important field of study.

The modern commercial smartphones are normally equipped with many useful features of sensors for scientific research, including but not limited to: GPS, accelerometer, gyroscope, touch, and proximity, which provide us with various approaches for driver phone use detection. We broadly categorize related works into three groups: special devices based detection, applications based detection, and smartphone sensors based detection. More specifically, some approaches based on special devices to detect driver distractions have been investigated [2, 8, 19, 20, 22], but these approaches heavily rely on the assistance of external devices. Some applications devoting to mitigating driver phone distractions have been developed [5, 23, 24, 29, 34, 35, 38, 41], such as *QC-Hold* [29], and *Negotiator* [38]. Nevertheless, these studies either require prior knowledge of phone use by the driver or blindly block calls/text of all the phones inside the vehicle. Some methods of built-in smartphone sensors dedicating to driver distractions have been studied, such as methods of using accelerations and cellular signal strength [3, 9, 11, 26, 36], and using smartphone embedded sensors to alert dangerous driving, monitor road conditions, and detect traffic accidents [6, 13, 18, 27, 39]. However, these solutions do not provide an option for drivers to determine whether to entitle phone control.

To eliminate driver distractions, we present *SafeDrive*, a driver phone use determining system that automatically determines the driving conditions leveraging the built-in smartphone sensors and then makes a flexible control of driver phone use while driving, in this paper. More specifically, we first explore GPS and accelerometer sensors on an Android Samsung Galaxy S4 smartphones to collect data from a real road driving vehicle, which can sufficiently capture driving conditions. With inputs of these data, we provide an accurate driving condition classification algorithm, that classifies driving conditions into five categories: {Local, Idle}, {Local, Busy}, {Highway, Idle}, {Highway, Busy}, and {Traffic Jam}. Based on the classified driving condition, *SafeDrive* makes a flexible control of driver phone use while driving. Finally, we excessively evaluate the classification accuracy of our *SafeDrive* in local, highway, traffic jam, and complex conditions, respectively, and the results demonstrate that it can achieve up to 87% classification accuracy in complex conditions.

The main contributions of this work can be summarized as follows:

- We present a driver phone use determining system, *SafeDrive*, that automatically determines the driving conditions leveraging the built-in smartphone sensors and then makes a flexible control of driver phone use while driving.
- We provide an accurate driving condition classification algorithm, that classifies driving conditions into five categories, with inputs of GPS and accelerometer sensor data.
- We excessively evaluate the classification accuracy of our *SafeDrive* in local, highway, traffic jam, and complex conditions, respectively, and the results demonstrate that it can achieve up to 87 % classification accuracy in complex conditions.

The remainder of this paper is organized as follows: Section 2 presents related works on driver's dangerous behavior detection. We describe the detailed *SafeDrive* architecture including sensor data collection, classification algorithm and phone call determination in Section 3 and evaluate the accuracy of *SafeDrive* in different driving conditions in Section 4. In Section 5, we discuss advantages and disadvantages including limitations and latency issues of *SafeDrive* in the experiments and introduce the future work, and conclude the paper in Section 6.

2 Related work

There are tremendous works focusing on detecting driver's dangerous behavior while driving, which can be broadly categorized into three groups: special device based detection, application based detection, and smartphone sensors based detection.

2.1 Special devices based detection

Some approaches based on special devices to detect driver distractions have been investigated [2, 8, 19, 20, 22]. The authors of [2] present a non-intrusive prototype computer vision system for monitoring a driver's vigilance in real time, which is based on a hardware system for the real-time acquisition of a driver's images using an active IR illuminator and the software implementation for monitoring some visual behaviors that characterize a driver's level of vigilance. In [19], the authors propose a facility for monitoring the distraction of a driver, which is able to detect the driver's visual and cognitive workload by fusing stereo vision and lane tracking data, running both rule-based and support-vector machine (SVM) classification methods. In [20], *Key2SafeDriving* requires special devices installed inside the vehicle to enable blocking cellular communications of a special phone based on the readings from the vehicle's speedometer or even rely on a radio jammer [8]. The authors of [22] explore the potential for wearable devices to identify driving activities and unsafe driving, without relying on information or sensors in the vehicle.

However, these approaches heavily rely on the assistance of external devices, which increase the cost of the implementation and reduce the portability of the detection system.

2.2 Applications based detection

Some applications devoting to mitigating driver phone distractions have been developed [23, 24, 29, 38, 41]. In [29], the authors propose *QC-Hold*, a Quiet Calls prototype, that combines three buttons for responding to calls with a PDA/mobile phone unit to silently send pre-recorded audio directly into the phone. The authors of [38] provide *Negotiator* that embodies three main design requirements: support for negotiation, contextual information about when a recipient is available for a call, and lightweightness to reduce attention overhead. In [23], the authors present *BlindSight*, a prototype application that users interact using the phone keypad, without looking at the screen, which responds with auditory feedback. The authors of [24] propose using context-awareness to implement burden-shifting, time-shifting, and activity-based sharing to mitigate the problem of distracted driving caused by mobile phones. In [41], the authors propose a machine-learning-based method for detecting driver cell phone usage using a camera system directed at the vehicles front windshield.

Some applications focusing on blocking incoming or outgoing calls and texts for mobile phones have been investigated [5, 34, 35]. In [5], *DriveSmart* can automatically direct incoming calls to voicemail, defer messages and other data transactions and alert a parent when overridden while driving. In [34], *Textecution* is designed for parents to install on their teenage driver's phone so they know their child is safer behind the wheel of the vehicle. In [35], *tXtBlocker* is the leading supplier of solutions to prevent distracted while driving incidents caused by using a mobile device while operating a motor vehicle.

Nevertheless, these studies either require prior knowledge of phone use by the driver or blindly block calls/text of all the phones inside the vehicle. Our *SafeDrive* only blocks incoming calls when the driver is under critical driving conditions such as on a highway or on a busy local way, which uses data from the phone sensors to perform driving condition classification.

2.3 Smartphone sensors based detection

Some built-in smartphone sensors dedicating to driver distractions have been studied. Some methods using accelerations and cellular signal strength [3, 9, 11, 26, 36] to detect the moving vehicle have been developed. In [9], the authors indicate a potentially large improvement using UMTS signalling data compared with GSM regarding handover location accuracy, where these improvements can be used to generate real-time traffic information with higher quality and extend the geographic usage area for cellular-based travel time estimation systems. The authors of [26] propose *PEIR*, the Personal Environmental Impact Report, which is a participatory sensing application that uses location data sampled from everyday mobile phones to calculate personalized estimates of environmental impact and exposure. In [36], the authors describe a crowd-sourced alternative to official transit tracking, which we call cooperative transit tracking. The authors of [3] consider the problem of tracking fine-grained speeds variations of vehicles using signal strength traces from GSM enabled phones. In [11], the authors design an app on iPhone for reducing the smartphone-related distracted driving, which can run in the background and can lock the smartphone screen with no passwords required when it detects that the user is driving.

Other studies use smartphone embedded sensor to alert dangerous driving, monitor road conditions, and detect traffic accidents [6, 13, 18, 27, 39]. The authors of [27] present *Nericell*, that performs rich sensing by piggybacking on smartphones that users carry with them in normal course. In [6], the authors propose a highly efficient system aimed at early detection and alert of dangerous vehicle maneuvers typically related to drunk driving, which requires only a mobile phone placed in vehicle and with accelerometer and orientation sensor. The authors of [18] propose a novel system that uses Dynamic Time Warping (DTW) and smartphone based sensor-fusion (accelerometer, gyroscope, magnetometer, GPS, video) to detect, recognize and record these actions without external processing. In [39], the authors automatically detect traffic accidents using accelerometers and acoustic data, immediately notifies a central emergency dispatch server after an accident, and provides situational awareness through photographs, GPS coordinates, VOIP communication channels, and accident data recording. The authors of [13] provide an analysis of iPhone's `CurrentPowerlog.powerlogsystem` file and Android device buffer logs, along with their associated residual data, both of which can potentially be used to establish mobile phone usage at the time of, or leading up to, a motor vehicle accident.

Smartphone sensors used to monitor road conditions have been investigated [7, 14, 15, 28, 32]. The authors of [7] describe a system and associate algorithms to monitor this important civil infrastructure using a collection of sensor-equipped vehicles. In [28], the authors describe a mobile sensing system for road irregularity detection using Android OS based smart-phones. The authors of [32] present a vision based method to automatically determine if a driver is holding a cell phone close to one of his/her ears (thus keeping only one hand on the steering wheel) and quantitatively demonstrate the methods efficacy on challenging Strategic Highway Research Program (SHRP2) face view videos from the head pose validation data that was acquired to monitor driver head pose variation under naturalistic driving conditions. In [14], the authors present *SmartRoad*, a crowd-sourced road sensing system that detects and identifies traffic regulators, traffic lights, and stop signs. The authors of [15] utilize smartphone sensors to estimate the vehicle speed, especially when GPS is unavailable or inaccurate in urban environments. In particular, they estimate the vehicle speed by integrating the accelerometer's readings over time and find the acceleration errors can lead to large deviations between the estimated speed and the real one.

Smartphone sensors used to detect driver phone have been studied [4, 40, 42, 43]. In [4], the authors present a driver detection system (DDS) by utilizing multiple sensors (including accelerometer, gyroscope, and microphone) in smartphones to capture the features of driver? movement. However, this approach is sensitive to the behavior of each individual, and highly depends on the position where drivers carry the phone, which is less practical. The authors of [42, 43] introduce an acoustic relative-ranging system that classifies on which car seat a phone is being used leveraging the car? audio infrastructure, which relies on the vehicle? audio system. In [40], the authors utilize smartphone sensing of vehicle dynamics to determine driver phone use, which can facilitate many traffic safety applications.

Different from these works, our *SafeDrive* outperforms for two reasons: 1) it provides finer-grained classification for driving conditions, which means in certain cases (for example, on an idle local way), the incoming calls are still allowed; 2) it gives user options of whether they allow it to take control of their phones.

3 System design

To eliminate driver distractions, we present a driver phone use determining system, *SafeDrive*, that automatically determines the driving conditions leveraging the built-in smartphone sensors and then makes a flexible control of driver phone use while driving. In this section, we first present the application requirements, and then describe our *SafeDrive* hardware and application. Next, we describe the *SafeDrive* architecture in detail.

3.1 Application requirements

Our driver phone use determining system is motivated by the requirements of a safe drive application for driver-passenger classification [40, 42, 43]. Data from multiple built-in sensors are transmitted to a smartphone processor which makes classification decisions in real time. The system is able to efficiently and accurately classify typical road conditions including road type and road status, which are tabulated in Table 1.

Table 1 *SafeDrive* classification categories

Road type	Road status
Local	Idle
Local	Busy
Highway	Idle
Highway	Busy
H/L	Traffic Jam

As illustrated in Table 1, we divide the driving conditions into five categories, where road types are divided into `Highway` and `Local`, and road status into `Idle`, `Busy`, and `Traffic Jam`. To this point, we actually have six combinations of the driving conditions. However, `Traffic Jam` can happen on road types of `Highway` and `Local`. Since the behavior of local traffic jam or highway traffic jam is actually similar with very slow speed and intermittent stops, we merge these two kinds of road types into one as `Traffic Jam`. In addition, we believe the category of `Traffic Jam` is indispensable, because drivers are prone to checking their phones for a longer waiting in serve traffic jam driving condition. Such road condition classification is quite useful for driver determining and safe driving applications. The requirements to provide such a helpful driving condition determining system are: user-friendly, accurate and efficient classification, and less reliance on ground truth.

- **User-friendly.** The hardware must be portable and lightweight, while the software must provide an intuitive interface for adding, removing, and configuring different sensors geared to detect the driver intended conditions.
- **Accurate and efficient classification.** The system must be able to handle both easy and difficult to detect road conditions, and adapt the variable orientation of the smartphones. The system must be energy efficient since a smartphone usually is powered by energy-constrained batteries.
- **Less reliance on ground truth.** Classification systems are often deployed with minimal labeled training data, therefore, the need to train a system online requests ground truth labels only when absolutely necessary. A reduced request of ground truth reduces the burden on the user to label training data.

3.2 *SafeDrive* hardware and application

To achieve accurate and efficient classification for built-in smartphone sensors, we provide a description of hardware and software used in our system. In this section, we first introduce the hardware used in the system, and then describe the graphical user interface (GUI).

3.2.1 Hardware description

The modern commercial smartphones are equipped with many useful features of sensors for scientific research, including but not limited to: GPS, 3-axis accelerometer, 3-axis gyroscope, touch, and proximity. These devices are powerful, inexpensive and versatile research platforms that make data collection accessible to the general public. Our *SafeDrive* system is solely deployed on a Samsung Galaxy S4 smartphone with Android operating system 4.2.2

(Jelly Bean), and sensors of GPS, accelerometer, gyroscope, gesture, and proximity. In particular, the reason we choose GPS to navigate *SafeDrive* is that GPS is often preferred over its alternatives such as GSM/WiFi based positioning systems since it is known to be more accurate [16, 30, 33]. It is well known that GPS is power hungry, but it is not a key issue for drivers' smartphones because they can be always charged by their vehicles or mobile power banks.

3.2.2 Android application

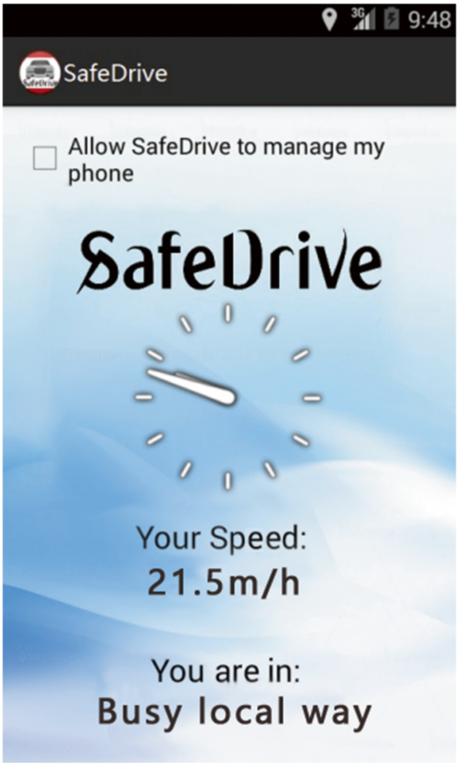
To provide a user-friendly front end for *SafeDrive*, we implement an Android application to allow for vehicle speed evaluation, driving condition detection, current time illustration, data storage and upload for analysis, as illustrated in Fig. 1. The GUI provides an option for driver to decide whether to entitle our system phone control. That is, the GUI allows phones to receive incoming calls when the drivers are in a relatively good driving condition [1].

3.3 *SafeDrive* architecture

The basic idea of our *SafeDrive* system is to eliminate driver distractions from phones when they are in critical driving conditions. In this section, we first describe the *SafeDrive* system overview in general, and then describe the core of our *SafeDrive* platform in detail.

Our *SafeDrive* system resides solely on an Android smartphone with no reliance on a backed server. Multiple built-in sensors on phones such as GPS and accelerometer

Fig. 1 *SafeDrive* GUI



☐ Allow SafeDrive to manage my phone

Your Speed:
21.5m/h

You are in:
Busy local way

sample data at a default configured rate for road status classification. We choose a configured time window to collect data. Then the data are logged on the phone locally and the types of stored data are `(time stamp, vehicle speed, acceleration)`, where $acceleration$ is a 3-dimensional field with elements of x-, y-, and z-axis values. Next, the data of configured time interval are fed into the *SafeDrive* classification system at an interval to make a classification decision. Further, based on the obtained driving conditions, the *SafeDrive* system can automatically manage the user phones as long as the users have allowed the application to take control (a box to check is provided in *SafeDrive*). A typical and distinct scenario of the *SafeDrive* system is that if the user is driving in a busy highway, the phone will block all incoming phone calls to help the driver stay focused. We now describe the core of our *SafeDrive* system:

3.3.1 Sensor data collection

We introduce how we collect the speed data from GPS and how we use the data from x, y, z axes of acceleration.

GPS data Android operating system supports a location class, in which a set of APIs are provided. The location class encapsulates the speed function, where the speed is calculated from the distance of two GPS locations (such as latitude and longitude) over a time interval. We exploit the `location.getSpeed()` function to obtain the current vehicle speed with the default sampling rate 1 Hz [1]. We do not tune this rate since it is the fastest rate that works in most of phones, and we desire high accuracy [17].

Accelerometer data We directly read the values of x, y, and z axes from the accelerometer. The three dimensions are defined as shown in Fig. 2, where the x and y axes are parallel to the phone screen, and the z-axis is perpendicular to the screen. However, the gravity force is always influencing the device, so the acceleration can be calculated as $-g - \frac{\sum F}{mass}$. Therefore, the magnitude of acceleration is $9.81m/s^2$ when the phone is still. An essential problem is that the orientation of phone impacts the real values because of 3-dimensional

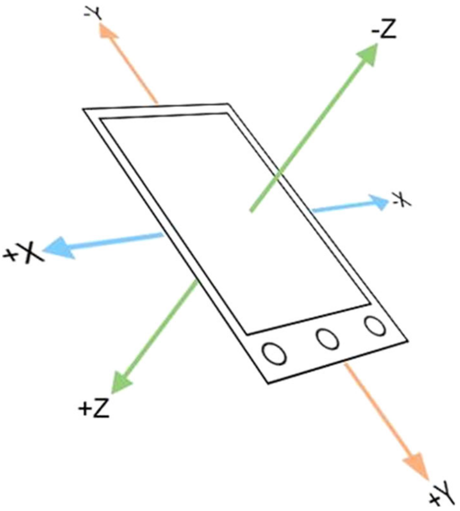


Fig. 2 Dimensions of accelerometer

acceleration. In order to obtain the true values of the acceleration, we need to find a solution to eliminate the effect of gravity, even though the 3-dimensional acceleration is difficult to decompose. We provide two potential solutions: 1) a low-pass filter, which is able to isolate the force of gravity, and then take out the gravity force along the 3 axes; 2) a fixed orientation, that is, put the phone at a certain orientation. Suggested by Android web site [1], we implement the low-pass filter solution, however, we find that it not only filters out the gravity force, but also filters out much useful and valuable information, such as turning and slight brake. The reason is likely to be that vehicles usually have slow accelerations under normal operations, which are prone to be filtered out. As for the solution of fixed orientation, it is lightweight and cheap for the phone to do gravity decomposition. Therefore, we choose the fixed orientation of phones to eliminate the effect of gravity in our *SafeDrive* system, as illustrated in Fig. 3, where the smartphone screen is parallel to the ground, facing upwards, and the head points forward. In the experiment, we place the phone horizontally with head pointing forwards, which adds the gravity force solely on the z axis. We discard the z-axis data since this dimension of acceleration does not really work on our driving condition classification.

3.3.2 Classification algorithm

As illustrated in Table 1, we divide the driving conditions into five combinations: {`Local, Idle`}, {`Local, Busy`}, {`Highway, Idle`}, {`Highway, Busy`}, and {`Traffic Jam`}. We describe the classification algorithm in Algorithm 1 and with a flowchart of Fig. 4, respectively. Using Algorithm 1, we classify driving conditions into $\{type, status\}$ with inputs of speed from GPS and y-axis values from accelerometer $(speed, Y\_acceleration)$. Illustrated in Algorithm 1, we first configure a time window of 30 seconds for data sampling, that is, $t = 30$. We use this 30-second data including $speed$ and $Y\_acceleration$ to do the classification. Then, we count the peak number of $Y\_acceleration$ and assign the corresponding value to pa, and calculate the highest speed by $max(speed)$. Next, we compute the ratio of the number of $speed$ that is less than 5 mph over the number of $speed$, which is a threshold to classify the driving conditions. With these results, if $ratio_i > 0.2$ and $pa > 8$, then we classify road status as `Traffic Jam`; otherwise, if $(v_{max} > 55)$, we classify road type as `Highway`, or if $(v_{max} \leq 55)$, classify it as `Local`; if $pa > 5$, we classify road status as `Busy`, and if $pa \leq 5$, classify road status as `Idle`. With the classified $\{type, status\}$, *SafeDrive* will determinate wether to alert drivers or block the phone directly, which we discuss in the following section.

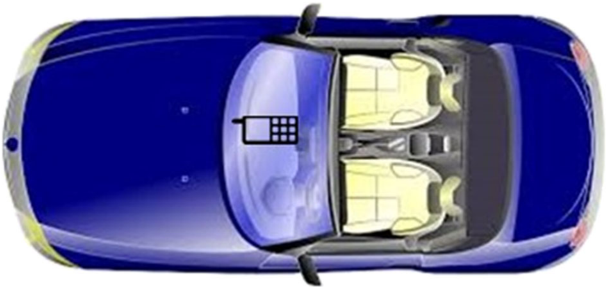


Fig. 3 Device orientation

Algorithm 1 The classification algorithm

```
Input: The speed from GPS and $y$-axis values from accelerometer:
       $(speed, Y\_acceleration)$
Output: The road type and status: $(type, status)$
$DetermineType(speed[t], Y\_acceleration[t])$ // "t" denotes the data sampling time interval, which is assigned 30 seconds in the implementation.
$pa = countpeak(Y\_accleration)$ // "pa" indicates the number of peaks in y axis of acceleration.
$v_{max} = max(speed)$ //max() is used to calculate the highest speed.
$ratio_i = \frac{count(speed<=5)}{count(speed)}$ // "i" represents the ith classification of driving conditions.
if $ratio_i > 0.2$ and $pa > 8$ then
    $status = Traffic\ Jam$
    else
        if $(v_{max} > 55)$ then
            $type = Highway$;
            else
                $type = Local$;
            end
        end
        if $pa > 5$ then
            $status = Busy$;
            else
                $status = Idle$;
            end
        end
    end
end
Return $\{type, status\}$
```

3.3.3 Phone use determination

We aim to protect drivers from phone distractions by blocking incoming phone calls under critical conditions or alerting drivers under good conditions. In App level, *SafeDrive* uses the `TelephonyManager` class and its method listen to register a listener `PhoneStateListener` that will detect `PHONE_State_CHANGED` notified by `TelephoneRegistry` in Framework level. If call state is changed from `CALL_State_IDLE` to `CALL_State_RINGING`, *SafeDrive* will determine whether block the phone call or not based on the classification. We define three driving conditions as critical conditions: {`Highway, Busy`}, {`Highway, Idle`}, and {`Local, Busy`}. Once driver's smartphone is classified that the driver is under any one of the three conditions, our *SafeDrive* on the smartphone automatically blocks any incoming calls. On the other hand, once the driver is under the other two conditions {`Local, Idle`} and {`Traffic Jam`}, *SafeDrive* believes the driver is in a safe and good driving condition and it will alert the driver, so that phone calls can come in.

In addition, *SafeDrive* respects the driver's will. If a driver does not feel comfortable letting *SafeDrive* manage the phone call, the driver can uncheck the box on the GUI, as shown in Fig. 1, thereby disabling *SafeDrive*'s managing mechanism. On the GUI, *SafeDrive* still reports some useful information, such as current speed and current driving condition, without *SafeDrive*'s management.

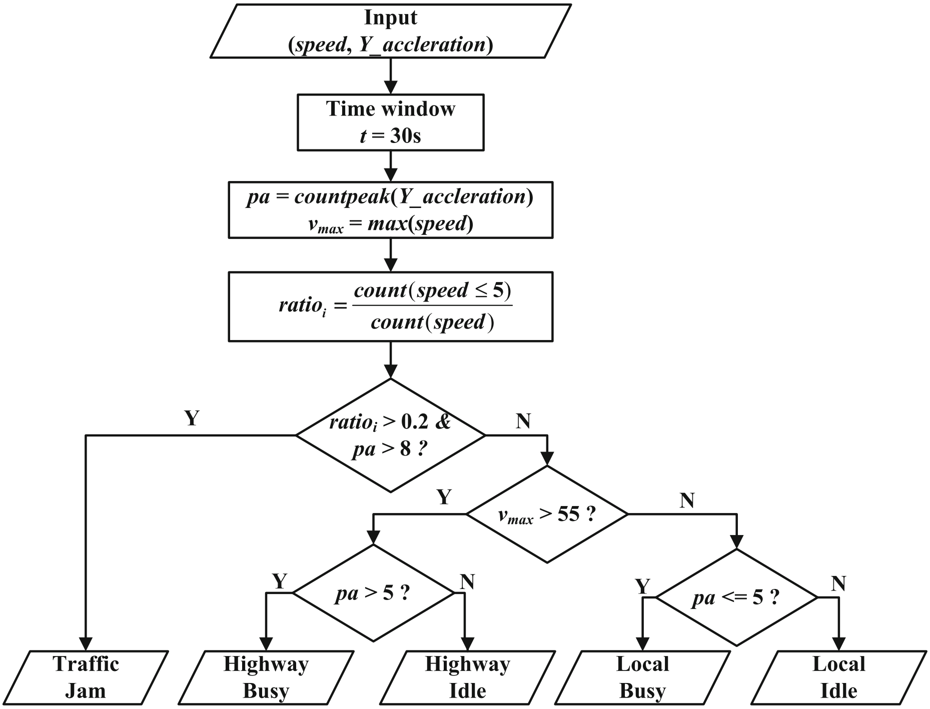


Fig. 4 Flow of the classification algorithm

4 Performance evaluation

We evaluate *SafeDrive* with real driving tests. In this section, we first describe our evaluation setup, then present accuracy evaluation of *SafeDrive* in different driving conditions, and finally compare *SafeDrive* with other related systems.

4.1 Evaluation setup

In the experiments, we solely implement *SafeDrive* on a Samsung Galaxy S4 smartphone, and place the smartphone on the middle of the dashboard of a vehicle. As described in Section 3.3.1 and Fig. 3, the smartphone is mounted in a fixed orientation so that we can directly retrieve data from different axes. Then, we drive the car in different conditions for about two hours, and obtain data from GPS and accelerometer. In order to perform quantitative analysis, *SafeDrive* automatically logs its readings from GPS and accelerometer into the smartphone's internal storage.

In order to evaluate *SafeDrive*'s accuracy in vehicle speed measurement and driving condition classification, we need to record real vehicle speed and driving conditions. To achieve this, we maintain synchronized video logs while *SafeDrive* is collecting data. In the video logs, we record readings from the vehicle's speedometer, which serve as ground truth for driving speed.

4.2 Accuracy evaluation

In this section, we evaluate the accuracy of *SafeDrive* in terms of speed, acceleration, and different driving conditions including local, highway, traffic jam, and complex conditions, respectively.

4.2.1 Speed measurement

In order to evaluate the accuracy of vehicle speed measurement by *SafeDrive*, we compare speed values from *SafeDrive* with the corresponding ground truth. Since *SafeDrive* updates its vehicle speed measurement every second, then we conduct the comparison every second. We randomly choose 2-minute data from *SafeDrive* and video logs, respectively, for comparison. We calculate the accuracy of vehicle speed measurement by (1).

$$Accuracy_{speed} = 1 - \frac{\sum |c - s|}{\sum s} \times 100\ \% \tag{1}$$

where "*c*" represents the obtained speed from GPS and "*s*" represents the readout of speedometer from video logs. Based on the calculation, the accuracy is up to 98 % over our 2-minute GPS data.

In addition, since the vehicle's speedometer only displays a digital readout in format of integer, we define that *SafeDrive* is reliable if the difference between *SafeDrive* measurement and ground truth is within 0.5 mph. For instance, if *SafeDrive* updates the measurement with a readout of 45.8345 mph, while vehicle speedometer (or ground truth) displays 46 mph at that time point, we say *SafeDrive* measurement is reliable. On the other hand, if the vehicle speedometer displays 45 mph, we say this measurement is unreliable. Under this definition, the overall reliability is very high, and among those, we detect only 9 unreliable cases. Therefore, we can say that the speed data from GPS are fairly accurate, and are sufficiently used as inputs of classification algorithm.

4.2.2 Acceleration measurement

We start with a 6-minute raw data of local conditions from y axis of the acceleration to evaluate acceleration accuracy.

With the selected raw data, we plot the y axis of acceleration in m/s^2 and the corresponding vehicle speed in mph, where the vehicle runs on {`Local, Busy`} and then turns to {`Local,Idle`}, and the speed reaches 50 mph but does not exceed 55 mph. As illustrated in Fig. 5, x axes correspond to elapsed time, while the y axis of the upper subfigure represents acceleration, and that of lower subfigure indicates the corresponding vehicle speed. From the upper subfigure, we can see that signals from accelerometer are more noisy, partially due to its higher sample rates and partially due to its high sensitivity. In real experiment, from 60th second to 130th second, the vehicle runs on a straight road, and then experiences two sudden brakes because of unexpected driving conditions. These are reflected in two negative spikes in acceleration and the corresponding changes in speed. More over, at around 150th second, the vehicle first slows down and stops at a signal light for around 10 seconds; then turns right, and enters a quiet and idle road. We can see variations in acceleration are much smaller, and the speed is mostly stable.

From the analysis, we deduce that changes in accelerations reflect the frequency of applying brakes and speeding up, which are indicators of the road traffic. That is to say,

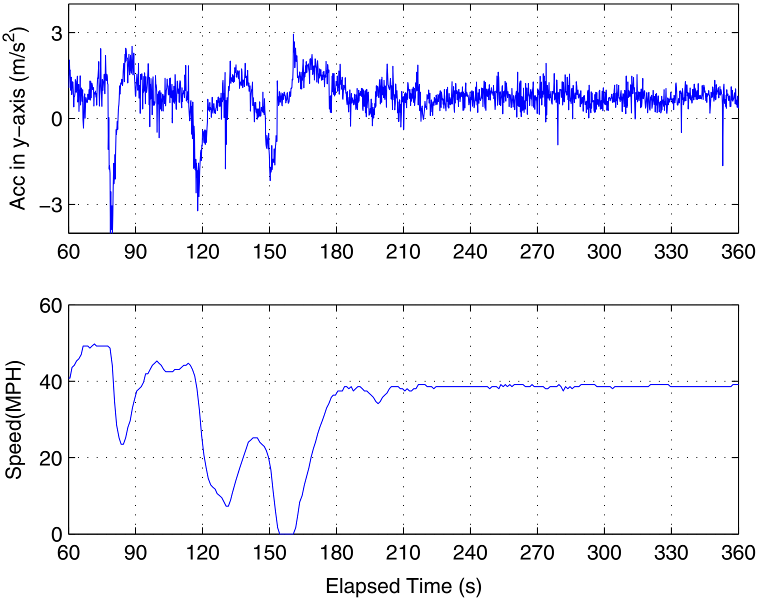


Fig. 5 Example of local conditions (raw data)

there are more spikes in acceleration if there is much traffic. As a result, we use the number of peaks to differentiate busy driving conditions from idle. *Median Filtering* is used to perform noise reduction, which runs through data and replaces each data point by the median of all neighboring data points. In *SafeDrive*, we explore zero padding to address boundary issue and the size of sliding window is 10, which works very well in filtering out noise while preserving real information. *Peak-Finding* technique is used to detect positive and negative peaks in acceleration, which conventionally locates the maximum of every three data points. In *SafeDrive*, we make two modifications that: 1) set a threshold when detecting peaks for reducing false positives; 2) penalize peaks that are very close to each other to avoid duplicated peaks. Based on Fig. 5, we plot the filtered acceleration signal with detected peaks in the upper subfigure of Fig. 6, where yellow lines represent raw acceleration, blue lines indicate the filtered acceleration signal, red circles indicate positive peaks, and black circles indicate negative peaks. From this subfigure, we can clearly see that there are 3 negative peaks associated with 3 brakes and 12 positive peaks from 60th second to 180th second. In contrast, there are only 3 positive peaks and no negative peak starting from 120th second. This phenomenon perfectly matches our intuition that when there is a lot of traffic, the vehicle will speed up or slow down more frequently. In the lower subfigure, green " + " represents {`Local, Idle`} and red " + " represents {`Local, Busy`}. Note that the reason that positive peaks after 120th second do not lead to a noticeable increase in speed is that *Median Filtering* performs the noise reduction.

4.2.3 Different driving conditions

Driving conditions can be classified into 5 groups as illustrated in Table 1. In this section, we present and analyze *SafeDrive*'s accuracy and efficiency of classification in local, highway,

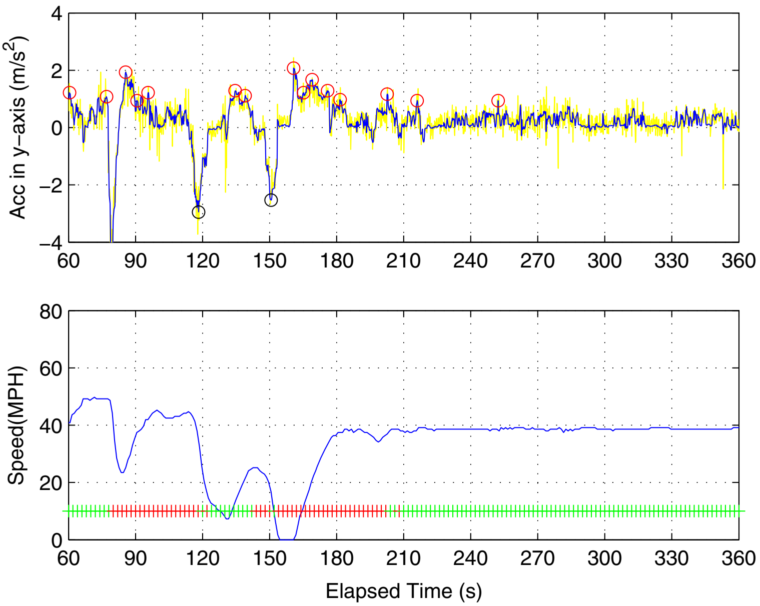


Fig. 6 Example of local conditions (data processing and classification)

traffic jam, and complex conditions, respectively. As introduced in Algorithm 1, *SafeDrive* takes the speed and acceleration in the past 30 seconds as input, and then classifies driving conditions into one of the five groups.

Local conditions As with previous example, *SafeDrive* conducts 150 estimations and the results are marked as the lower subfigure of Fig. 6, where " + " indicates `type = Local`, meaning that driving condition is estimated as "Local", and red " + " denotes `status = Busy` while green " + " indicates `type = Idle`. The comparison of *SafeDrive*'s estimations and ground truth is listed in Table 2, where numbers in the second and third columns indicate time ranges that *SafeDrive*'s estimations or ground truth fall into the corresponding driving conditions. For instance, from 78th second to 118th second, and 142nd second to 202nd second, our *SafeDrive* classify the driving condition into {`Local`,`Busy`}, while the ground truth is that from 80th second to 212nd second the driving condition is {`Local`,`Busy`}. For other time intervals, the driving condition is {`Local`,`Idle`}. Based on the comparison, 132 out of the 150 estimations are correct, which indicates that our *SafeDrive* has 88% classification accuracy for local conditions.

Highway conditions We then discuss the accuracy of highway condition classification. We plot the obtained data in Fig. 7. In the experiment, the vehicle run steadily on highway

Table 2 Results of local conditions example

Driving conditions	SafeDrive	Ground truth
{`Local`,`Busy`}	78–118, 142–202	80–212
{`Local`,`Idle`}	Others	Others

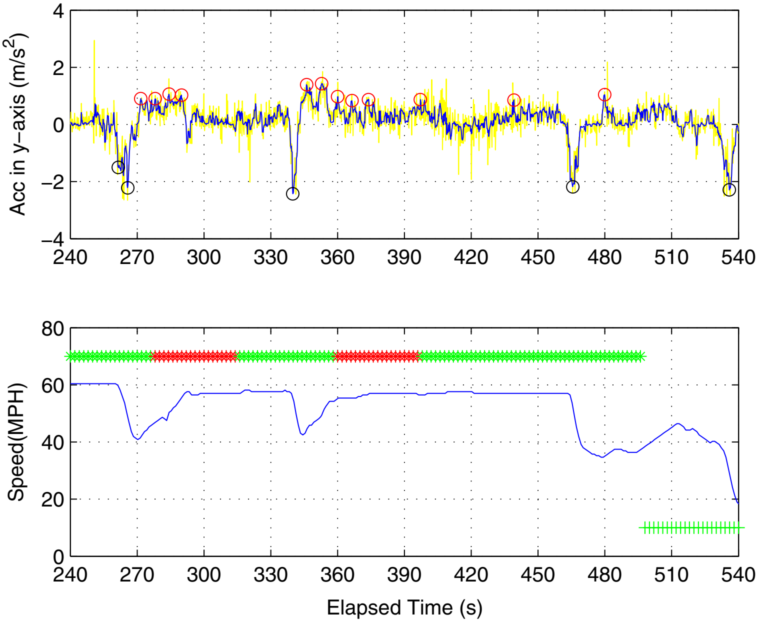


Fig. 7 Example of highway conditions

except for two cases where brakes apply, at 270th and 344th seconds, respectively, and then the vehicle slows down and exits the highway at around 466th second. Our *SafeDrive* conducts the estimations, and the corresponding results are illustrated on the lower subfigure of Fig. 7, where " + " indicates `type = Highway`, meaning that driving condition is estimated as "Highway", and green " + " indicates {`Highway,Idle`} while red " + " represents {`Highway,Busy`}. The comparison of our *SafeDrive*'s estimations and ground truth is summarized in Table 3. As illustrated in Table 3, 96 out of the 150 estimations are correct, which demonstrates that the *SafeDrive* has 64 % accuracy of classification for highway conditions. Among those missed cases, 19 of them are caused by the latency when the vehicle exits the highway, which we will discuss in Section 5.

Traffic jam conditions We next discuss a scenario where we experience a traffic jam as shown in Fig. 8. The estimations made by our *SafeDrive* is plotted on the lower subfigure, where red " + " represents {`Local,Busy`} while black " △ " indicates {`Traffic Jam`}. The results are summarized in Table 4, where 121 out 150 cases are correct, which demonstrates that our *SafeDrive* has 81 % accuracy of classification for traffic jam conditions.

Table 3 Results of highway conditions example

Driving conditions	*SafeDrive*	Ground truth
{`Highway,Idle`}	Others	240–262, 380 – 464
{`Local,Idle`}	496–540	466–540
{`Highway,Busy`}	276–314, 360–394	264 – 378

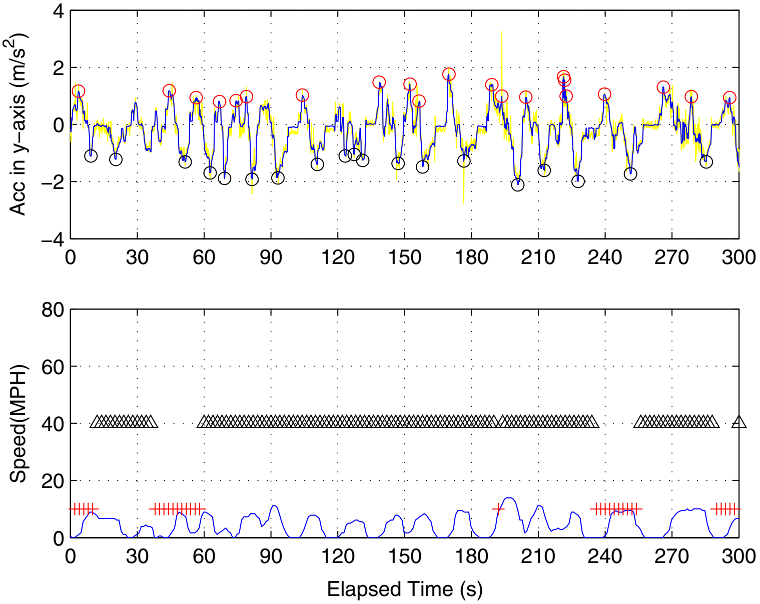


Fig. 8 Example of traffic jam

Complex conditions In order to show the overall accuracy of our *SafeDrive* under any conditions, we obtain 56-minute data by driving our vehicle in the experiment. With these data, we conduct 1695 estimations made by the *SafeDrive*, and the corresponding results are listed in Table 5, where the second column represents the total number of *SafeDrive*'s estimations, the third denotes the number of correct cases, and the last indicates the corresponding accuracy. From the table, our *SafeDrive* can correctly classify the driving conditions in 1472 cases, and achieve the overall accuracy up to 87 %. The overall accuracy is higher than that of single condition we illustrated. The reason is that when driving conditions are in transition, the accuracy is reasonably lower. In fact, driving conditions are probably in transition, our *SafeDrive* can effectively and accurately classify the driving conditions, and we also believe that the overall accuracy can even be much higher if the driving conditions do not vary so much.

4.3 Comparison with other systems

In order to make a comparison, we incorporate *Google Map Navigation* and *Waze* into *SafeDrive* on two Samsung Galaxy S4 smartphones to determine driver phone use, respectively, where *Google Map Navigation* is a turn-by-turn navigation application, and *Waze* is a community-based traffic and navigation application. To avoid confusion, we simply refer to

Table 4 Results of traffic jam example

Driving conditions	*SafeDrive*	Ground truth
{Traffic Jam}	2–8, 26–58, 192, 236–254, 288–298	All
{Local,Busy}	Others	None

Table 5 Results of overall evaluation

Driving conditions	Estimations	Correct cases	Accuracy
{Local,Busy}	433	388	90 %
{Local,Idle}	446	395	89 %
{Highway,Busy}	340	303	89 %
{Highway,Idle}	267	217	81 %
{Traffic Jam}	209	169	81 %
Overall	1695	1472	87 %

the two incorporated driver phone use determination systems as *GMNDrive* and *WazeDrive*, respectively. In the experiment, we vertically place the three smartphones on the middle dashboard of a vehicle. With the same settings to *SafeDrive*, we evaluate the accuracy of *GMNDrive*, *WazeDrive* and *SafeDrive* under any conditions by obtaining 30-minute data with real vehicle driving, respectively. With these data, we conduct 895 estimations for the three systems, and the corresponding comparison results are illustrated in Table 6. As illustrated in Table 6, the correctly classified cases by *GMNDrive*, *WazeDrive* and *SafeDrive* are 792, 780 and 782, respectively, and the overall accuracies are 88 %, 87 % and 87 %, respectively. Moreover, *WazeDrive* and *SafeDrive* have high accuracies in local road types, while *GMNDrive* have high accuracy in highway road types. This is probably because these applications are designed for different navigation environments. However, *GMNDrive* and *WazeDrive* require more smartphone storage for the applications and need an Internet connection, which greatly consumes the battery.

5 Discussion and future work

In this section, we first discuss advantages and disadvantages including limitations and latency issues of our *SafeDrive*, and then introduce the future work.

5.1 Discussion

Our *SafeDrive* can determine driver phone use leveraging built-in smartphone sensors sensing driving conditions, while the related systems such as *GMNDrive* and *WazeDrive* leverage live traffic maps and need Internet connection, which require more smartphone

Table 6 Comparison results

Driving	All	*GMNDrive*		*WazeDrive*		*SafeDrive*	
Conditions	Estimations	Correct Cases	Accuracy	Correct Cases	Accuracy	Correct Cases	Accuracy
{Local,Busy}	212	186	88 %	191	90 %	190	90 %
{Local,Idle}	225	193	86 %	205	91 %	201	89 %
{Highway,Busy}	162	147	91 %	137	85 %	142	88 %
{Highway,Idle}	159	147	92 %	133	84 %	134	84 %
{Traffic Jam}	137	119	87 %	114	83 %	115	84 %
Overall	895	792	88 %	780	87 %	782	87 %

storage and battery energy. Our *SafeDrive* prototype classifies driving conditions based on GPS and accelerometers with a classification algorithm, and a *SafeDrive* product can significantly reduce vehicle accidents.

We have observed a few weak predictions in our performance evaluation of the *SafeDrive*, which is largely due to two factors: information limitation and latency.

Limitations We do not consider history information, so we have unexpected standing out wrong predictions. For example, in Fig. 6, our results show that the driving condition is {`Local,Idle`} between 120th second and 142nd second. However, the ground truth is different. It is reasonable to expect driving conditions to be continuous, that is, driving conditions should be smoothly changing instead of jumping back and forth. Considering history information can help control such wrong predictions, however, we have not implemented more complex algorithm in *SafeDrive* currently. Although there are some wrong predictions standing out, it is likely that the drivers will not receive a call during a short duration. Note that the probability of incoming phone calls while driving follows the uniform distribution in statistics. Therefore, we believe the *SafeDrive* can still be very useful even with such a limitation. In addition, we ignore the turning information in our *SafeDrive* because it is too weak to observe. Under highly noisy environment, it is quite hard to identify a turn when a driver changes a lane or makes a slight turn for the accelerometer, and only sharp turns can be clearly captured in our experiments. Therefore, we discard this type of information, using only y-axis acceleration information.

Latency There is sometimes a latency in our estimations. One example is that the vehicle has exited the highway at the 466th second, however, *SafeDrive* did not change the road status to {`Local,Idle`} until the 496th second, as shown in Fig. 7. This is because the inputs of our *SafeDrive* are previous 30-second data, which are used to determine current driving conditions. So the number of peaks, maximum speed within a sliding window (30 seconds) in the algorithm cannot be changed promptly as we promptly enter another driving condition. Therefore, there would be a gap between our estimation and the ground truth. Nevertheless, the latency is a short time of a few seconds, which does not influence a long period of driving condition classification.

5.2 Future work

There can be much to do based on our current system. First, we can add more functions on the protection mechanism of *SafeDrive*, such as delaying SMS and notifications from other applications. We can also make customized blocking plan or provide blacklist and white list. Second, we can investigate a better solution to eliminate the gravity force on the accelerometer. We currently place our phones at a fixed position to benefit classification. This may need to be further improved to increase driver experience. Finally, we can reduce or even eliminate the use of GPS since it is quite energy hungry. We may take advantage of accelerometer to calculate the speed using integration of acceleration [25, 31], and may consider vehicle board computers. In addition, we may consider how driving styles of drivers affect the performance of *SafeDrive*.

6 Conclusion

As mobile devices have been widely used in human daily life, people start using them in inappropriate cases, even when they are driving. It has been common sense that using a mobile phone while driving is dangerous and numerous reports have proved the threat of driver distractions. Even law has banned certain usage of mobile phones in some states in US. Although being informed of the potential danger, people still keep using their phones while driving.

To eliminate driver distractions, we present *SafeDrive*, a driver phone use determining system that automatically determines the driving conditions leveraging the built-in smartphone sensors and then makes a flexible control of driver phone use while driving. More specifically, we first explore GPS and accelerometer sensors on an Android Samsung Galaxy S4 smartphones to collect data from a real road driving vehicle, which can sufficiently capture driving conditions. With inputs of these data, we provide an accurate driving condition classification algorithm, that classifies driving conditions into five categories: {Local, Idle}, {Local, Busy}, {Highway, Idle}, {Highway, Busy}, and {Traffic Jam}. Based on the classified driving condition, *SafeDrive* makes a flexible control of driver phone use while driving. Finally, we excessively evaluate the classification accuracy of our *SafeDrive* in local, highway, traffic jam, and complex conditions, respectively, and the results demonstrate that it can achieve up to 87 % classification accuracy in complex conditions.

Acknowledgments We would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China (Grant nos. 61528206, 61402380, and 61402063), the Natural Science Foundation of CQ CSTC (Grant nos. cstc2015jcyjA40044 and cstc2013kjrcqnrc40013), U.S. National Science Foundation (Grant nos. CNS-1253506 (CAREER) and CNS-1250180), the Fundamental Research Funds for the Central Universities (Grant no. XDJK2015B030), the State Ethnic Affairs Commission of China (Grant no. 14GZZ012), and the Science and Technology Foundation of Guizhou (Grant no. LH20147386).

References

1. Android Developer, http://developer.android.com/reference/android/location/Location.html
2. Bergasa L, Nuevo J, Sotelo M, Barea R, Lopez M (2006) Real-time system for mointoring driver vigilance. IEEE Trans Intell Trans Syst 7(1):63–77
3. Chandrasekaran G, Tam V, Varshavsky A, Gruteser et al (2011) Tracking vehicular speed variations by warping mobile phone signal strengths. In: 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom'11), pp 213–221
4. Chu H, Raman V, Shen J, Choudhury R, Kansal A, Bahl V (2011) Poster: You driving? talk to you later. In: ACM 9th Annual International Conference on Mobile Systems, Applications, and Services(MobiSys'11), pp 1–1
5. Drivesmart plus. http://support.t-mobile.com/docs/DOC-2374
6. Dai J, Teng J, Bai X, Shen Z, Xuan D (2010) Mobile phone based drunk driving detection. In: 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), vol 10, pp 1–8
7. Eriksson J, Girod L, Hull B, Newton R, Madden S, Balakrishnan H (2008) The pothole patrol: Using a mobile sensor network for road surface monitoring. In: Proceedings of the 6th annual international conference on Mobile systems, applications, and services (MobiSys'08), pp 29–39
8. Guardian angel vehicle platform. http://www.trinitynoble.com/
9. Gundlegard D, Karlsson J (2009) Handover location accuracy for travel time estimation in GSM and UMTS. IET Intell Transp Syst 3(1):87–94
10. Governers Highway Safety Associaltion. http://www.ghsa.org/html/stateinfo/laws/cellphone_laws.html

11. Gao X, Tian J, Wang G (2014) Poster: detection of transportation mode based on smartphones for reducing distracted driving. In: Proceedings of the 20th annual international conference on Mobile computing and networking (MobiCom '14), pp 355–358
12. Horrey WJ, Wickens CD (2006) Examining the impact of cell phone conversations on driving using meta-analytic techniques. Human Factors 48(1):196–205
13. Horsman G, Conniss LR (2015) Investigating evidence of mobile phone usage by drivers in road traffic accidents. In: Proceedings of the Second Annual DFRWS Europe (DFRWS 15 Europe), pp 30–37
14. Hu S, Su L, Liu H, Wang H, Abdelzaher TF (2015) Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. ACM Trans Sensor Netw (TOSN) 11 (4):1–27
15. Han H, Yu J, Zhu H, Chen Y, Yang J, Zhu Y, Xue G, Li M (2014) Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments. In: Proceedings of the 33rd Annual IEEE International Conference on Computer Communications (INFOCOM'14), pp 727–735
16. Hu S, Su L, Li S, Wang S, Pan C, Gu S, Amin MTA, Liu H, Nath S, Choudhury RR, Abdelzaher TF (2015) Experiences with eNav: A low-power vehicular navigation system. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'), vol 15, pp 1–12
17. Hedgecock W, Maroti M, Sallai J, Volgyesi P, Ledeczi A (2013) High-accuracy differential tracking of low-cost GPS receivers. In: Proceedings of the 11th annual international conference on Mobile systems, applications, and services (MobiSys'13), pp 221–234
18. Johnson D, Trivedi M (2011) Driving style recognition using a smartphone as a sensor platform. In: Proceedings of the 14th IEEE International Conference on Intelligent Transportation Systems (ITSC'11), pp 1609–1615
19. Kutila M, Jokela M, Markkula G, Rue M (2007) Driver distraction detection with a camera vision system. In: IEEE International conference on Image Processing (ICIP'07), pp 201–204
20. Key2safedriving app. http://www.key2safedriving.com/
21. Laberge-Nadeau C, Maagb U, Bellavanceb F et al (2003) Wireless telephones and the risk of road crashes. Accid Anal Prev 35(5):649–660
22. Liu L, Karatas C, Li H, Tan S, Gruteser M, Yang J, Chen Y, Martin RP (2015) Toward Detection of Unsafe Driving with Wearables. In: Proceedings of the 2015 workshop on Wearable Systems and Applications (WearSys'), vol 15, pp 27–32
23. Li KA, Baudisch P, Hinckley K (2008) BlindSight: Eyes-free access to mobile phones. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08), pp 1389–1398
24. Lindqvist J, Hong J (2011) Undistracted driving: A mobile phone that doesn? distract. In: Proceedings of the 12th Workshop on Mobile Computing Systems and Applications (HotMobile'11), pp 70–75
25. Li Y, Qi X, Ren Z, Zhou G, Xiao D, Deng S (2011) Energy Modeling and Optimization through Joint Packet Size Analysis of BSN and WiFi Networks. In: Proceedings of 30th International Performance Computing and Communications Conference (IPCCC'), vol 11, pp 1–8
26. Mun M et al (2009) PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In: Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys'09), pp 55–68
27. Mohan P, Padmanabhan V, Ramjee R (2008) Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In: Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys'08), pp 323–336
28. Mednis A, Strazdins G, Zviedris R, Kanonirs G, Selavo L (2011) Real time pothole detection using android smartphones with accelerometers
29. Nelson L, Bly S, Sokoler T (2001) Quiet calls: Talking silently on mobile phones. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01), pp 174–181
30. Paek J, Kim J, Govindan R (2010) Energy-efficient rate-adaptive GPS-based positioning for smartphones
31. Qi X, Keally M, Zhou G, Li Y, Ren Z (2013) AdaSense: Adapting Sampling Rates for Activity Recognition in Body Sensor Networks. In: Proceedings of 19th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'), vol 13, pp 163–172

32. Seshadri K, Xu FJ, Pal DK, Savvides M, Thor CP (2015) Driver Cell Phone Usage Detection on Strategic Highway Research Program (SHRP2) Face View Videos. In: Proceedings of the 6th international Workshop on Computer Vision in Vehicle Technology (CVPR'), vol 15, pp 35–43
33. Saremi F, Fatemieh O, Ahmadi H, Wang H, Abdelzaher T, Ganti R, Liu H, Hu S, Li S, Su L (2015) Experiences with greengps—fuel-efficient navigation using participatory sensing. IEEE Transactions on Mobile Computing (TMC). doi:10.1109/TMC.2015.2421939
34. Textecution. http://www.textecution.com/
35. Txtblocker. http://www.txtblocker.com/
36. Thiagarajan A, Biagioni J, Gerlich T, Eriksson J (2010) Cooperative transit tracking using smart-phones. In: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10), pp 85–98
37. US Department of Transportation - National Highway Traffic Safety Administration (2014). Distracted Driving 2012. Traffic Safety Facts Research Note, 1-8. http://www-nrd.nhtsa.dot.gov/Pubs/812012.pdf
38. Wiberg M, Whittaker S (2005) Managing availability: Supporting lightweight negotiations to handle interruptions. ACM Trans Comput-Human Interac 12(4):356–387
39. White J, Thompson C, Turner H, Dougherty B, Schmidt D. C (2011) WreckWatch: Automatic traffic accident detection and notification with smartphones. Mob Netw Appl 16:285–303
40. Wang Y, Yang J, Liu HB, Chen YY, Gruteser M, Martin R. P (2013) Sensing vehicle dynamics for determing driver phone use. In: Proceedings of the 11th annual international conference on Mobile systems, applications, and services (MobiSys'13), pp 41–54
41. Xu B., Loce R. P. (2015) A machine learning approach for detecting cell phone usage. In: Proceedings of Video Surveillance and Transportation Imaging Applications (SPIE '15). doi:10.1117/12.2083126
42. Yang J, Sidhom S, Chandrasekaran G, Vu T, Liu H et al (2011) Detecting driver phone use leveraging car speakers. In: Proceedings of the 17th annual international conference on Mobile computing and networking (MobiCom'11), pp 97–108
43. Yang J, Sidhom S, Chandrasekaran G, Vu T, Liu H, al et. (2012) Sensing driver phone use with acoustic ranging through car speakers. IEEE Trans Mob Comput 11(9):1426–1440

Yantao Li is an Associate Professor in the College of Computer and Information Sciences at Southwest University, China. He received the PhD degree from the College of Computer Science at Chongqing University, in December 2012. From September 2010 to September 2012, he was a visiting scholar supported by China Scholarship Council working with professor Gang Zhou at the Department of Computer Science in the College of William and Mary, USA. His research area includes wireless communication and networking, sensor networks and ubiquitous computing, and information security.

Gang Zhou is an Associate Professor in the Department of Computer Science at the College of William and Mary. He received his Ph.D. degree from the University of Virginia in 2007 under Professor John A. Stankovic. He has published more than 60 papers in the areas of sensor networks, ubiquitous computing, mobile computing and wireless networks. The total citations of his papers are more than 4300 according to Google Scholar, among which the MobiSys'04 paper has been cited more than 780 times. He also has 13 papers each of which has been cited more than 100 times since 2004. He is an Editor of IEEE Internet of Things Journal and also an Editor of Elsevier Computer Networks Journal. Dr. Zhou served as NSF, NIH, and GENI proposal review panelists multiple times. He also received an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008. He is a recipient of the Best Paper Award of IEEE ICNP 2010. He received NSF CAREER Award in 2013. He is a Senior Member of IEEE and a Senior Member of ACM.

Yue Li is working toward the PhD degree in the Department of Computer Science at the College of William and Mary.

Du Shen is working toward the PhD degree in the Department of Computer Science at the College of William and Mary.