Detection of Distracted Driving:
A Smartphone-Based Approach

Giuseppe Cancello Tortora, Mirko Casini, Andrea Lagna, Martina Marino,
and Alessio Vecchio(✉)

University of Pisa, Pisa 56122, Italy
`alessio.vecchio@unipi.it`

Abstract. Road accidents cause 1.35 million deaths a year and have now become the leading cause of death between the ages of 5 and 29. CAReful is an application that can be used while driving to detect dangerous behaviors, such as drowsiness, turning the head, using the smartphone, or the presence of excessive noise. CAReful uses different sensors to monitor the driver, such as the microphone, GPS, camera, accelerometer, gyroscope, and magnetometer, and to obtain information about the vehicle's speed and the road. Since driving behavior is privacy-related, all processing and storage of sensitive information occur on the user's device.

Keywords: Road accidents · Distraction detection · Smartphone

1 Introduction

According to the "Global status report on road safety", the number of road traffic deaths is approximately 1.35 million every year, and road accidents are the main cause of death for people aged 5–29 years [18]. A significant fraction of road accidents is known to be caused by distracted driving. The U.S. Department of Transportation's National Highway Traffic Safety Administration (NHTSA) estimates that the number of deaths caused by distraction was +3300 in 2011, whereas the number of injuries was 387000 [15]. The problem is known to be relevant also in other parts of the world [8], and in more recent years the situation did not get better [16].

Distraction can be defined as a specific type of inattention where the driver focuses on non-driving activities. Distractions can be categorized in *visual distractions*, such as looking at the infotainment system of the vehicle, *manual distractions*, such as eating or drinking, and *cognitive distractions*, such as speaking with other passengers [17]. In many cases, a distracting task can span more categories: for instance, reading a text message on a smartphone requires both looking away from the road and using at least one of the hands to control the smartphone. Distractions can also be more or less demanding in terms of cognitive load.

The increasing capabilities of vehicles, in terms of automated driving, can be helpful in the long term, as the driving tasks will be shifted from humans

to the vehicle's intelligence. However, the level of automation that is currently available, or that will be available in the near future, can be paradoxically an incentive to distraction [12]. In fact, some levels of automated driving require the driver to be attentive and promptly take control of the vehicle in anomalous situations, but being the human relieved most of the time from driving tasks, distracting activities like using a smartphone can occur even more frequently. In addition, the time needed to replace all existing vehicles with fully automated ones is going to take several years.

As a consequence, methods for automatically detecting distracted driving received significant attention during the last years. In this paper, we propose a system for detecting distracted driving that advances the state of the art along the following directions: i) the method identifies a wide range of distracting activities relying on the multiple sensors available on a smartphone (the camera, the microphone, the GPS, the IMU); ii) detected distractions are used to compute a distraction index that can be used by the driver for self-assessing their driving style; iii) the system relies only on a smartphone and all computation and storage of information occur locally. This is important because information about the driving style is of personal nature and it could be used for malevolous purposes (e.g. by insurance companies not authorized by the user). Finally, in some countries, monitoring of workers is not allowed; therefore, a solution like the one we propose can be useful for those professional drivers who are interested in self-assessing their driving style but, at the same time, want to preserve their privacy.

2 Related Work

Because of the social significance of the problem, the detection of distracted drivers received significant attention during the last few years. Several approaches are based on computer vision only. Through computer vision, in fact, it is possible to detect a wide range of distracting behaviors, such as using a smartphone while driving, reaching out objects in the backseat, or looking in the wrong direction. In [6], an approach based on two stages is presented: the first stage relies on a ResNet-101 network to identify and locate the relevant elements in an image, such as hands, face, and a smartphone; the second stage uses features like the distance between the previously identified elements to classify the current behavior as safe or not. The approach was evaluated on a set of images when executed on a standard PC. Another vision-based approach is described in [14], where a camera is used to recognize a set of distracting activities such as texting using the phone, operating the radio, doing makeup, reaching behind, or talking with other passengers. The focus of the study was on the best performance that can be obtained using a number of well-known deep learning methods in terms of accuracy and execution speed on embedded computers. The problem of optimizing a Convolutional Neural Network (CNN) is faced in [3], where the number of parameters is reduced compared to other CNN-based approaches. Also in this case, the approach is based only on images

of the driver to recognize the unsafe behaviors included in the dataset provided
in [7].

Detection of unsafe behaviors that is specifically designed for being executed
on smartphones is described in [19]. Unsafe behaviors are not restricted to inat-
tentive driving, but also include careless change of lane, tailgating another vehi-
cle, and lane weaving. One of the main goals of the system is to select the most
appropriate camera depending on the current context. The reason is due to the
inability of the adopted smartphones to activate the front-facing camera when,
at the same time, the rear one is in use. The front-facing camera is used to rec-
ognize the possible drowsiness of the driver or the direction of the head, whereas
the other camera is needed to observe the trajectory of the car across lanes and
the distance from the preceding vehicle. The approach focuses on a wide range of
danger sources, but does not specifically target the possibly different distracted
driving behaviors.

SafeDrive is a wearable system able to detect a number of distracting actions
[11]. SafeDrive relies on the IMU available on smartwatches to monitor the right-
hand movements of the driver, to understand if she is interacting for too much
time with the car controls, she is searching items located at the passenger's seat
or at the backseat, or she is eating. The approach uses, as the main source of
information, the rotation angle of the driver's hand on the horizontal plane,
integrated with information collected by the smartphone. The latter, in partic-
ular, is useful to monitor the vehicle's dynamics. Another approach based on
wearable devices is the one described in [13], where the position of the smart-
watch is derived using the RSSI of Bluetooth Low Energy (BLE) communica-
tion. The basic idea is that the smartwatch communicates with the passengers'
smartphones so that its position can be estimated without relying on IMUs (the
rationale is that IMUs are also influenced by vehicle dynamics). The set of dis-
tracting actions again includes using a smartphone, eating or drinking, searching
onboard items, and operating the vehicle systems. IMUs were used in [9], where
the considered distracting activities were all related to the use of a smartphone
(being involved in a call, two-way texting, and reading a message).

The reader is forwarded to [12], where existing literature about the detection
of driver's distraction is reviewed and taxonomized.

3   CAReful: An App for Detecting Distracting Behaviors

CAReful is an Android application for smartphones able to measure the level of
inattention of the user while driving. The app makes use of the different sensors
typically available on a smartphone to collect information about the behaviour
of the driver and about the environment around her (e.g., tortuosity of the road).
The distracting behaviors that CAReful is able to detect are: drowsiness, turned
head, usage of smartphone, smartphone fall, and excessive noise. The device is
supposed to be positioned in front of the user as it is commonly done when
using a smartphone as a navigation aid. The main sensors used by CAReful
are: the GPS, the microphone, the camera and all the motion-related sensors,

such as the accelerometer, the gyroscope and the magnetometer. Beside the distracting behaviors, CAReful combines the presence of such activities with the vehicle speed and the tortuosity of the travelled road. The idea is that distracting behaviors are much more dangerous as the speed increases or when the road is non-rectilinear. The output is a single index that can be used to evaluate the general risk associated to distractions. The index is computed during a trip and saved to keep note of the user's behavior. In the following, we describe how the considered distracting activities are detected.

3.1   Drowsiness

The front camera is used for the detection of the driver drowsiness. In particular, drowsiness is recognized by computing the fraction of time the eyes of the driver are closed. The camera sampling rate is set 10 Hz, as a trade-off between detection accuracy and resource consumption. To recognize the features of the driver's face, we rely on the Google ML Kit [10]. In particular, we make use of the Face Detection APIs to detect the points of interest within a face and then to estimate the fraction of time the eyes are kept open or closed. Whenever the driver's eyes are kept closed for 20 consecutive frames ($\sim$2 s), then a drowsiness counter is incremented. Such a counter is then combined with similar ones concerning the other possibly distracting activities to obtain the final index.

3.2   Turned Head

The same camera is used also to understand if the driver has turned her head to the left or to the right, looking away from the road. Also in this case, we relied on the Face Detection APIs as it is highly optimized for being executed on mobile devices. The driver's head is considered to be turned when the module of the Euler Y angle of the head is greater than a threshold (40°). The threshold has been set considering that when the Euler Y angle is greater that 36°, only the right eye is visible from the camera. The same applies when the angle is less than −36°. Once the driver has turned her head in one of the two directions for at least 4 s, the counter related to this distracting behavior is increased.

3.3   Usage of Smarthphone

The IMU of the smartphone provides information useful to detect if the device is used by the driver. In particular, we adopted a simple approach based on detecting a rotation of the device, corresponding to when the smartphones is removed from the holder where it is supposed to be placed. The listener of the gyroscope is activated with a period of $\sim$200 ms. If the module of the angular velocity becomes greater than the threshold, the distracting activity is detected.

3.4   Smartphone Fall

We decided to include the fall of the smartphone in the set of distracting activities because picking up the device from the vehicle floor can be extremely dangerous. To detect a fall, the magnitude of the acceleration vector is computed and stored in a buffer. We decided to use the magnitude of the acceleration vector, and not all the three components, to keep the method simple and independent from the possible rotations of the device when falling [1,2,5]. The magnitude values in the buffer are then compared with two thresholds – *lowTh* and *highTh* – to detect the free fall of the smartphone and then its impact onto the floor. Every time a new value is added to the circular buffer, the algorithm searches in the array for a value lower than *lowTh*, corresponding to the free fall phase. Then, if the free fall phase is detected, the algorithm searches for a value higher than *highTh* in the remaining part, caused by the impact of the device onto the floor.

3.5   Excessive Noise

The microphone of the smartphone is used to capture the level of noise in the vehicle. The *android.media AudioRecord* class is used to sample the sound level every 500 ms, encoded as 16 bit PCM. The mean value of the samples is computed to make the system tolerant to possible short spikes in the signal. The mean value is then compared to a reference value to obtain the level of noise on a dB scale. The reference value used is the minimum value that the device is able to measure. Unfortunately, the result cannot be easily translated into a dB Sound Pressure Level[1] (SPL) because the latter assumes a reference level of $20\mu Pa$ and the samples are collected through an uncalibrated smartphone. For this reason, we finally compared the result to an empirically derived threshold. A study demonstrated that reasonably accurate measurements can be carried out when using iOS devices as the different models share many similarities from the point of view of the audio subsystem [4].

3.6   Trip Logging and Road Tortuosity

Every trip has departure and arrival coordinates, collected via GPS. The GPS is also used to retrieve the speed of the vehicle which is used as a multiplying factor when computing the distraction index, as discussed in Sect. 4. The rotation vector, derived from the accelerometer and magnetometer readings, is used to compute the curvature index of the road covered during the trip. The azimuth is sampled every 200 ms and the difference between adjacent samples is used as an indication of the direction change in the considered period. Finally, the average module of the differences is used as an indication of the overall road tortuosity. The result is then compared to a set of empirically-defined thresholds to obtain the curvature index.

---

[1] dB SPL is typically used to express the threshold of discomfort or pain for humans.

Table 1. Contribution of every distraction counter and multiplier in the score formula

	Role	Range	Description
Tortuosity	multiplier	1–3	Difficulty of the road
Speed	multiplier	1–4	Speed of the vehicle
Drowsiness	counter	0–30	How many times eyes are closed
Head turned	counter	0–15	How many times head is turned
Usage	counter	0–1	Phone is used
Fall	counter	0–1	Phone falls
Noise	counter	0–12	How many times noise is detected

4   Distraction Score

As mentioned, the distraction score quantifies the level of distraction of the driver at a given time during the trip. The distraction score d is computed according to the following formula:

$$d = (\frac{Speed}{4} + \frac{Tortuos.}{3}) \cdot (\frac{Head}{15} + \frac{Drowsiness}{30} + \frac{Noise}{12} + Fall + Usage) \quad (1)$$

Speed and tortuosity act as a multiplying factor. The other term, instead, expresses the amount of the driver's distracting behaviors. The speed index is supposed to be in the 1–4 range as we identified 4 different speed limits to be typically adopted (urban road, extra-urban road, main extra-urban road, and motorway). Road tortuosity is expressed by a value in the 1–3 range, as we wanted to classify roads into easy, medium, and difficult ones. The counters related to drowsiness, turned head, and smartphone usage are normalized in the range 0–1 by dividing their actual values by the theoretically maximum values obtainable in a minute. The range of possible values for the different counters is shown in Table 1. This is done to compute the fraction of time a specific behavior has been detected. Fall and usage of smartphones are considered binary values, as they are associated with particularly distracting behaviors. During the whole trip, the attention level is re-computed every minute (and the counters are reinitialized). This enables identifying specific parts of the trip that have been characterized by distractions.

Equation 1 is used to compute the "istantaneous" disattention level. However, to provide a more readable feedback to the driver, variations are smoothed through an exponential moving average:

$$d_i = (\alpha \cdot d) + (1 - \alpha) \cdot d_{i-1}$$

The value of d_i is then shown to the user using a range of colors as depicted in Fig. 1b. To be more specific, what is actually presented to the user is the attention level, as we feel that providing positive feedback to the driver is better than highlighting bad behaviors.
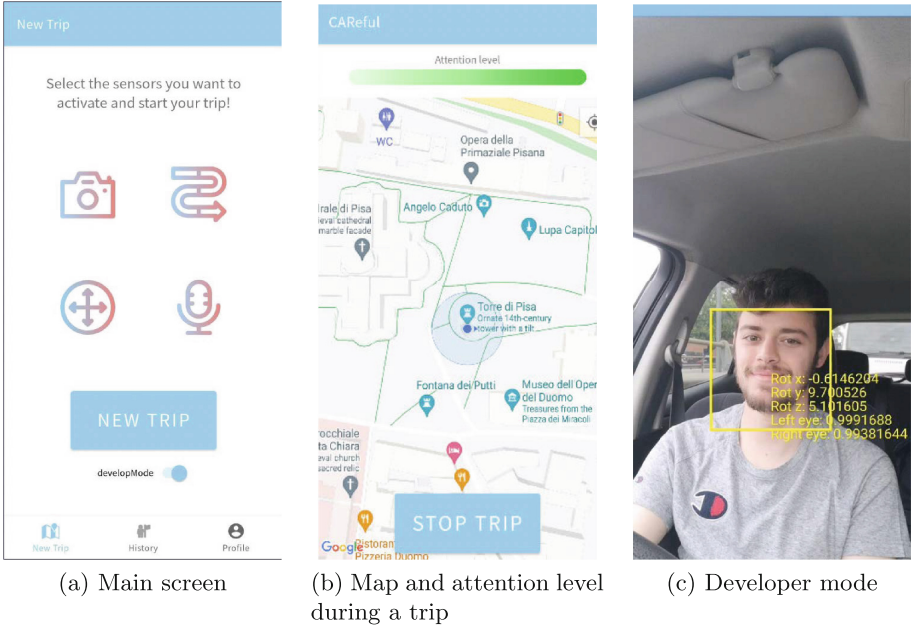
(a) Main screen   (b) Map and attention level during a trip   (c) Developer mode

Fig. 1. Screenshots of the app.

5   Conclusion

CAReful is an application that does not store any sensitive data, all the information collected about the driver and the surrounding environment is processed in real-time and not stored. Processing occurs locally using just the smartphone resources, and without transmitting any information to external services. The only information stored is a summary of the trip: departure and arrival location, departure and arrival time, and aggregate score. We believe that self-assessing the users' driving style while preserving their privacy is key to the widespread adoption of tools like the one we propose and, in the end, to improve safety on the roads.

Some screenshots of the app are shown in Fig. 1. In particular, Figs. 1a and 1b show the main screen and the map that is visualized when driving. Figure 1c instead shows the development mode of the app, where information about collected data and intermediate results is reported.

Future work will concern the evaluation of the app. As also pointed out by the literature summarized in Sect. 2, the evaluation of systems aimed at detecting distractions when driving is particularly troublesome because real tests are too risky. A possible solution is represented by realistic driving simulators, possibly integrated with a trace-driven execution as far as the motion sensors are concerned.

References

1. Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., Vecchio, A.: A smartphone-based fall detection system. Pervasive Mob. Comput. **8**(6), 883–899 (2012). https://doi.org/10.1016/j.pmcj.2012.08.003, https://www.sciencedirect.com/science/article/pii/S1574119212000983. special Issue on Pervasive Healthcare

2. Abbate, S., Avvenuti, M., Cola, G., Corsini, P., Light, J., Vecchio, A.: Recognition of false alarms in fall detection systems. In: 2011 IEEE Consumer Communications and Networking Conference (CCNC), pp. 23–28 (2011). https://doi.org/10.1109/CCNC.2011.5766464

3. Baheti, B., Talbar, S., Gajre, S.: Towards computationally efficient and realtime distracted driver detection with mobileVGG network. IEEE Trans. Intell. Veh. **5**(4), 565–574 (2020). https://doi.org/10.1109/TIV.2020.2995555

4. Celestina, M., Hrovat, J., Kardous, C.A.: Smartphone-based sound level measurement apps: evaluation of compliance with international sound level meter standards. Appl. Acoust., **139**, 119–128 (2018). https://doi.org/10.1016/j.apacoust.2018.04.011, https://www.sciencedirect.com/science/article/pii/S0003682X17309945

5. Cola, G., Vecchio, A., Avvenuti, M.: Improving the performance of fall detection systems through walk recognition. J. Ambient Intell. Humaniz. Comput. **5**(6), 843–855 (2014). https://doi.org/10.1007/s12652-014-0235-x

6. Dey, A.K., Goel, B., Chellappan, S.: Context-driven detection of distracted driving using images from in-car cameras. Internet Things, **14**, 100380 (2021). https://doi.org/10.1016/j.iot.2021.100380, https://www.sciencedirect.com/science/article/pii/S254266052100024X

7. Eraqi, H.M., Abouelnaga, Y., Saad, M.H., Moustafa, M.N.: Driver distraction identification with an ensemble of convolutional neural networks. J. Adv. Transp. **2019**, 4125865 (2019). https://doi.org/10.1155/2019/4125865

8. European Respiratory Society Observatory. European Commission. Road safety thematic report - driver distraction, D.G.f.T. (2022)

9. Goel, B., Dey, A.K., Bharti, P., Ahmed, K.B., Chellappan, S.: Detecting distracted driving using a wrist-worn wearable. In: 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 233–238 (2018). https://doi.org/10.1109/PERCOMW.2018.8480282

10. Google: Ml kit. https://developers.google.com/ml-kit. Accessed 15 Sept 2022

11. Jiang, L., Lin, X., Liu, X., Bi, C., Xing, G.: SafeDrive: detecting distracted driving behaviors using wrist-worn devices. Proc. ACM Interact. Mob. Wearable Ubiquit. Technol. **1**(4), 1–22 (2018). https://doi.org/10.1145/3161179

12. Kashevnik, A., Shchedrin, R., Kaiser, C., Stocker, A.: Driver distraction detection methods: a literature review and framework. IEEE Access **9**, 60063–60076 (2021). https://doi.org/10.1109/ACCESS.2021.3073599

13. Mewborne, T., Zhang, L., Tan, S.: A wearable-based distracted driving detection leveraging BLE. In: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, pp. 365–366. SenSys 2021, Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3485730.3492872

14. Tran, D., Manh Do, H., Sheng, W., Bai, H., Chowdhary, G.: Real-time detection of distracted driving based on deep learning. IET Intell. Transp. Syst. **12**(10), 1210–1219 (2018). https://doi.org/10.1049/iet-its.2018.5172, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2018.5172

15. Department of Transportation, N.H.T.S.A.: Traffic safety facts - research note - distracted driving 2011 (2011)
16. Department of Transportation, N.H.T.S.A.: Distracted driving (2020). https://www.nhtsa.gov/risky-driving/distracted-driving. Accessed 14 July 2022
17. Vegega, M., Jones, B., Monk, C., et al.: Understanding the effects of distracted driving and developing strategies to reduce resulting deaths and injuries: a report to congress. Technical report, United States. Office of Impaired Driving and Occupant Protection (2013)
18. (WHO), W.H.O.: Global status report on road safety (2018)
19. You, C.W.,et al.: CarSafe app: alerting drowsy and distracted drivers using dual cameras on smartphones. In: Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, pp. 13–26. MobiSys 2013, Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2462456.2465428