Eyes on the Road: Detecting Phone Usage by Drivers Using On-Device Cameras

Rushil Khurana
Carnegie Mellon University
Pittsburgh, PA, USA
rushil@cmu.edu

Mayank Goel
Carnegie Mellon University
Pittsburgh, PA, USA
mayankgoel@cmu.edu

ABSTRACT
Using a phone while driving is distracting and dangerous. It increases the accident chances by 400%. Several techniques have been proposed in the past to detect driver distraction due to phone usage. However, such techniques usually require instrumenting the user or the car with custom hardware. While detecting phone usage in the car can be done by using the phone's GPS, it is harder to identify whether the phone is used by the driver or one of the passengers. In this paper, we present a lightweight, software-only solution that uses the phone's camera to observe the car's interior geometry to distinguish phone position and orientation. We then use this information to distinguish between driver and passenger phone use. We collected data in 16 different cars with 33 different users and achieved an overall accuracy of 94% when the phone is held in hand and 92.2% when the phone is docked ($\leq$ 1 sec. delay). With just a software upgrade, this work can enable smartphones to proactively adapt to the user's context in the car and and substantially reduce distracted driving incidents.

CCS Concepts
•**Human-centered computing** → **Ubiquitous and mobile devices;**

Author Keywords
driver detection; position sensing; in-car behavior; situational impairments

INTRODUCTION
Smartphones are now ubiquitous and, over the years, their utility has increased exponentially. Being immensely pervasive and useful means users often choose to use their phones in dangerous situations. For example, 127 people have died between 2014 and 2016 while taking selfies on their phones [10]. While the users often ignore their safety, the phones are also unable to detect danger automatically. The phones do not adapt adequately to the user's situation and often contribute to making the situation more dangerous and amplify the associated risks.

ACM ISBN 978-1-4503-6708-0/20/04.
https://doi.org/10.1145/3313831.3376822

Figure 1. Lines detected in the photo captured by the phone when docked on the windshield at (a) passenger's right; (b) driver's left; and (c) driver's right side. The lines capture the perspective of the geometry of objects inside a car from different viewpoints.

One of the most common situations where phones increase the danger to their users is while driving. A 2014 survey conducted by National Highway Traffic Safety Administration (NHTSA) showed that 398 drivers were killed and 33,000 drivers were injured in accidents due to cell phone usage while driving [1]. Almost everyone knows that using a phone while driving is dangerous, but every time a notification pops up it demands attention. Driving creates a *situational impairment* [15] for the user, and the user's cognitive and visual focus is on their primary task – driving. To minimize driver distraction and improve safety, some apps (*e.g.,* Waze) disable the full set of functionalities while the car is in motion. It may be a suitable safety measure to deter individuals from using their phone while driving, but it is not enough. It asks the user if they are the driver or the passenger. If the user chooses to identify as a passenger, the full app functionality is regained. While this can be dangerous for a driver, blocking the entire phone can be counter-productive (especially because phones are widely used as navigation devices in cars). Therefore, we need the phones to sense and adapt to the user's context, *i.e.* driving, automatically.

Researchers have attempted to solve this challenge of detecting if the driver is using phones or in-car infotainment systems by instrumenting the car with electrodes [18, 5, 6], using a wearable on the driver's hand [12, 11] or inertial sensor data on the smartphone [3, 19, 4]. However, these approaches either rely on instrumentation of the car or user, or specific

event triggers (*e.g.,* detecting the direction in which the user opened the car door), which are not scalable solutions. An ideal system would identify if the driver were using their phone out-of-the-box without any modification or *in-situ* training. To this end, we present a lightweight, real-time, software-only solution that leverages the smartphone camera to determine if the driver or the passenger are using the phone. Given many users now use their phone's camera to unlock the phone, the camera is the perfect sensor to sense the usage context too. To build the system, we rely on the insight that irrespective of the car, the interiors of the cars are very similar. The exact placement, color, texture, *etc.* of the objects such as the handlebar, sunroof, visor, windows might change, but the basic geometry remains consistent. More specifically, when a driver uses the phone versus the passenger, it shows distinct perspectives of these shapes (geometries). We detect lines to capture this change in perspective as shown in Figure 1. The position and angle of these lines with respect to the position of the user's face provides enough information to train a robust machine learning model to distinguish between the driver and the passenger.

To develop and validate our machine learning models, we perform two studies: (1) in 10 different cars when the phone was docked to either the windshield or the air vents, and (2) 33 participants in 16 different cars to get different hand postures and approaches to hold the phone while driving (to ensure user safety the car was stationary). To elicit natural hand postures from the users, we asked the participants to pretend as if they were driving and using their phone. We demonstrate that within a second, our software-only approach can distinguish between the driver and the passenger with 95.5% accuracy when the phone is docked and 94% accuracy when the user holds the phone.

RELATED WORK

Sensing and countering driver distraction has been a long standing problem. There are several causes of distracted driving, but we focus our paper on driving and phone use. For such systems, there have been three distinct approaches in the past: (1) to instrument the car with minimal custom hardware; (2) to instrument the user (*e.g.,* wearables); and (3) using the smartphone itself to determine the user role. We look at each of these categories in the following sections.

Instrumenting the Car

To aid in detecting driver distraction, one of the most common and reliable approaches has been to leverage the car. The variety of electronics such speakers at every door or the infotainment system typically in the middle of the car have been previously used to predict the position of the person using a device.

Yang *et al*. leverage the acoustics in a car to infer the position of the phone in it [21]. They send a series of customized high-frequency beeps using the car stereo. They then use the time of arrival of the frequency back to the phone to estimate its position in the car with an accuracy of over 90%. While speakers are built into the car, the setup relies on a connection between the smartphone and the speakers. Secondly, the evaluation of this technique was in a controlled environment. The irregularity of human movement within the car may introduce multi-path interference that may cause the accuracy to diminish.

Another approach is to look inside the car using a camera. Drivers often place a dashcam that continuously records data about their driving; typically to mitigate insurance claims in case of an accident. Researchers have used an inward-looking dashcam to detect if the driver is using the phone or not. Berri *et al*. used a small dataset of 200 images to demonstrate that they can classify pictures of a person holding a phone with 87% accuracy in 3 seconds [2]. Similarly, Seshadri *et al*. used an existing dataset of a dashcam mounted on the windshield in the car to detect if the driver is making a call using their phone [16]. These solutions require instrumenting the car with external cameras, and they are only able to detect phone usage when its places near the ear. They are currently unable to solve the much larger problem of driving and texting.

Modern cars disable the touch systems on the infotainment system in cars when they are in motion. To make it an adaptive interface, researchers have investigated solutions that allow them to discriminate between the driver and passenger. Carpacio uses capacitive coupling to discriminate who is touching the screen of the infotainment system with an accuracy of 99.4% [18]. They inserted an electrode in each seat of the vehicle to measure the coupled signal between the capacitive screen and the electrode. But, they are not the first to use capacitive coupling to discriminate between users. Dietz *et al*. send a unique signal through the capacitive touchscreen of the device that is used by the electrode embedded in the seat to discriminate driver or passenger screen use [6]. Such systems are highly reliable but require custom hardware that makes it hard to deploy at scale.

Instrumenting the User

The proliferation of wearables provides a new *fixed* sensing point on the human body. Researchers have leveraged wearables to detect different activities, most notably to detect driving. WatchUDrive is a technique that uses the accelerometer and the camera on a smartwatch to distinguish between the driver and the passenger [12]. They note that holding a steering wheel is a restrictive activity and the inertial sensor might capture the repeatable pattern of motion and a camera on the smartwatch might capture a part of the steering wheel if the user is driving. They achieve an accuracy of 90% for a prediction within every 10 seconds using inertial sensors, whereas using just the camera, they were only able to obtain an accuracy of 62% within a 10 sec. window. Similarly, Liu et al. also used wrist-worn wearables to detect steering as a proxy to detect whose driving. But, their approach is limited to when a user is turning the car. The signal obtained by the inertial sensors has a unique signature when the user rotates the steering wheel. They evaluated their approach and achieved an accuracy of 98% [11]. Although the results are highly promising, the limited scope of the approach leaves room for improving the driver distraction systems. Furthermore, the accuracy of wearable solutions for activity recognition involving multiple limbs (*e.g.,* exercises [9], driving) is limited by their position

Figure 2. Different positions in which we collected the phone's camera data. The phone was docked in 3 positions on the windshield, 3 positions on the air vent, and used in the driver's hand and the passenger's hand. The users were free to switch hands as they preferred.

on the body. In this case, the wearable is unable to capture driving movements by the non-watch wearing hand.

Software-only Solutions

The most deployable systems are the ones that are self-contained, do not require custom hardware or elaborate setup. Several techniques have used the sensor suite present on a user's phone to determine if they are driving. Texive [3] is a software-only solution that relies on inertial sensor data to distinguish between the driver and the passenger with an accuracy of 87.18%. It uses IMU data to predict which side of the car did the person enter the car. It serves as a proxy to distinguish between the driver and passenger. Similarly, He *et al*. presented another system that relies on event triggers such vehicle turns and driving over an uneven road to discriminate patterns between the left and right side of the car with 90% accuracy [8]. But, their approach looks at relative changes in patterns of phones *i.e.,* an underlying assumption that two phones (users) are present in the car, and are accurately synchronized in time (<100 ms). Wang *et al*. overcome the issue of multiple phones, and suggest using an embedded accelerometer in a cigarette lighter adapter, or the OBD-II port adapter present in all cars [19]. Both approaches, however, either require custom hardware or interfacing with each car's OBD-II port, which makes it hard to deploy at scale.

Besides the IMU, the smartphone also has other sensors such as a microphone. Chu *et al*. used the fusion of audio and IMU sensors to identify micro-movements such as car entry, the direction of the action of wearing the seat belt, and the sound of turn signal sound to classify the position of the phone [4]. They were able to achieve an accuracy of 85% across six users in 2 different cars.

These approaches are based on event triggers, and failure to detect even one event can have an adverse cascading effect in determining if the driver is distracted. However, these approaches provide a groundwork for repeatable patterns one might observe in a car that we may be able to leverage. One approach may be to detect things like seat belt direction, the presence of a pedal, the position of the door *w.r.t.* the user. If a system can reliably detect these objects or similar patterns at any time, then we can eliminate the need for an event trigger, and build a real-time system.

DATA COLLECTION

In our data collection procedure, we have two variables:

1. ***Placement of the Phone***: *docked-shield*, *docked-vent*, *held-in-hand*
2. ***Camera Used***: *back-camera*, *front-camera*

In all conditions, the video was recorded at 30 frames per second with a resolution of 720p. The field of view of the camera is approximately 75 degrees.

Docked Phone

For the two docked conditions, we collected the data in 10 different cars[1]. We placed the phone in 6 different positions in the car, 3 each on the *shield* [Phone 1-3] and the *vent* [Phone 4-6] as shown in Figure 2. The positions were:

1. the left side of the driver facing towards the driver
2. the middle of the car faced towards the driver
3. the right side of the passenger faced towards passenger

When the phones were docked, the users did not need to interact with the phones. Thus, we did not recruit external participants for this part of the study. The members of the research team drove the cars in an urban area to collect the data. We chose this approach primarily because of the safety concerns around recording videos in a moving car. We recorded videos (avg. length = 3.5 mins.) from both the front and the back camera.

Phone in Hand

When the phone is held in the hand, apart from measuring the performance in different cars, we wanted to cover different user behaviors, postures, and approaches to holding the phone while driving. Thus, we recruited 33 participants (16 male, 17 female, mean age = 26.04) and recorded data in 16 different cars[2]. To ensure the safety of our participants, we conducted the study in a stationary car and simulated the in-hand conditions as shown in Figure 2 [phone 7-8]. We chose to conduct the study in a stationary car instead of a driving simulator to capture signals in a real setting and to capture visuals of real cars. When on the driver seat, the participants were asked to pretend as if they were driving and using the phone at the same time. They were encouraged to behave as they usually would while driving (eyes on the road, hands on the wheel *etc*.). Similarly, when the participants performed the task as a passenger, they were encouraged to behave/type as they would if they were passengers in a moving car. We did not control their phone usage behavior. The participants were allowed to move the phone or place the phone anywhere they desired. In fact some of them did place it in their lap, or the center console. This freedom allows us to capture more realistic data

[1]Ford Focus Hatchback, Toyota Prius, Honda Fit, Volkswagen Jetta, Ford Escape, Honda Odyssey, Ford Focus Sedan, Subaro Outback and Honda Civic '06, and Honda Civic '18

[2]Kia Rio, Subaru Outback '14, Subaru Outback '15, Honda Civic '06, Honda Civic '10, Mazda 3 '17, Mazda 3 '18, Toyota Corolla '10, Toyota Corolla '16, Prius '10, Prius '15, Prius '16, Nissan Rogue, Volkswagen Jetta, Toyota Camry, Ford Focus Hatchback

of phone usage in the car, instead of relying on predetermined positions chosen by us. The phone orientation was also not controlled, but all participants used the device in portrait mode while driving.

The participants completed two everyday tasks on their phone: (1) responding to text messages; and (2) changing music. These are the two most common tasks a person performs in their car that require continuous interaction. So, we used them as our study tasks to capture realistic scenarios. Both tasks were performed once as the driver and once as the passenger by the same person in their car. For the duration of the study, we recorded videos (avg length = 2.5 mins) from both the front and the back camera. These videos were recorded using an off-the-shelf app[3] that allows the phone to capture video while running in the background. This approach allowed the users to focus on their task and not get distracted by the video recording.

ALGORITHM

The goal of our work is to determine if the user of a phone is driver or passenger. A practical approach to such a problem needs to be lightweight, real-time, and immediately deployable. So, we built a software-only approach that phone makers can potentially push as a simple update.

We now discuss the underlying principle behind our approach. When a phone is used in a car, it is typically either in a person's hand or docked on a dock. Upon examining our data, we realized that the captured visuals look dramatically different for held and docked conditions, and hence would require separate models. Prior research has shown that inertial sensor data can be used distinguish if the phone is docked or held in hand [14, 7]. Despite not being a contribution of our work, for completeness, we built a model to verify that we could reliably do so. We used a Random Forest Classifier (default parameters, 10 trees) with the average delta in azimuth, pitch and roll (window=1s) from the phone's built in sensors, combined with the number of peaks from each of x, y and z axis of the accelerometer data (window=1s) as our features. We used the leave-one-car-out cross-validation to achieve an accuracy of 99.8%.

Next, we built a separate machine learning model for each of the two scenarios: *{docked, in-hand}*. We would like to point out that we used continuous video recording to obtain a large dataset of images. Our algorithm that uses the front camera runs on each individual frame and does not need continuous video for phone usage detection. To train both models, we balanced our data to contain equal instances of images of the driver and passenger. The algorithm for each situation is described in the following subsections.

Docked Phone

When a phone is docked on a vent or windshield, the front camera looks inwards into the car, and the observed scene shows that regardless of the car make and model, the interior looks very similar (Figure 1). Each car has windows, handlebars, seat belts, and sun visors. Depending on the car, the *exact* placement, color, and texture of these objects might vary, but the basic geometry of these objects remain consistent across cars.

Additionally, a phone at different positions in the same car has different perspectives, and the shape (geometry) of the objects seen inside the car varies by the phone's position. So, instead of relying on detecting different objects inside the car as a reference, we rely on detecting the shape and orientation of various objects in the scene. When a user looks at their phone in a car, the front camera captures many quadrilaterals (*e.g.,* handlebar, windows, visors). To encode the shape and orientation of these quadrilaterals, we rely on one of the most simple computer vision algorithms – detecting lines and used it to capture the change in perspective. To do so, in either of the docked positions, we extracted each frame from the recorded videos. We first recognize and locate the position of the user's face. We crop two regions of interest (ROIs): (1) above the face, and (2) under the face. We identify all possible lines in these two ROIs using the Probabilistic Hough Line Transform method. The lines above the face were used to extract the perspective of the quadrilaterals. The lines detected under the face were used to identify the orientation of the seat belt. For each of the lines, we first filter out the lines that intersect with each other. Next, we filter out the lines shorter than 10 pixels. Finally, we calculate the slope of each line using its leftmost point w.r.t. to the x-axis in the left-to-right direction. Given seat belts are always along the diagonal, we then filter the lines under the face with a slope between the ranges of $40°$ and $50°$ and $-40°$ and $-50°$.

Now, the number of lines detected in different frames may vary. So, for a frame, if n lines are detected, then we make n copies of that frame, each one representing a single line. We calculate the following features for each copy of the frame:

1. (x,y) coordinates of the leftmost point of the line
2. the angle of the line (in degrees) measured at the leftmost point *w.r.t.* to the left-to-right direction
3. (x,y) coordinates of the midpoint of the line
4. (x,y) coordinates of the centre of the bounding box that encapsulates the face of the user.

We use a Random Forest Classifier (max_depth = 16, 10 trees) to train our classifier with all copies of each frame as individual training instances. Then, to obtain a single output, we take a majority vote among all copies of the same frame. Finally, we take a majority vote of frames across a window of m second and vary m from 0 to 10.

From our data, we observed that the camera was often occluded when the phone is docked on the vent. While it worked in some cars, the view was entirely blocked by the vent design in others (as shown in Figure 3). But, when a phone is docked on the windshield, the back camera gets a completely unoccluded view of the outside of the car. Compared to the predictable observed geometry of various objects inside the car, the scene outside the car is dynamic and unpredictable. Thus, for the back camera, we use a different approach than detecting lines and their orientations. Prior research has used

[3]Background Video Recorder

Figure 3. Figure showing a limited view from the back camera of the smartphone when it is mounted on the air vent.

vanishing point detection from cameras looking outside the car as a tool to build driver assistance systems [17, 13]. We observed that different phone positions and orientations on the windshield lead to very different vanishing points as shown in Figure 4. Thus, we use the vanishing point as a feature. When the car is in motion, we look at the direction of the motion to determine the vanishing point. We start by extracting optical flow trajectories from our video using Lucas-Kanade sparse optical flow. The algorithm generates new keypoints every five frames and tracks them continuously across frames to produce a motion trajectory. A keypoint has a lifespan of 100 frames. A small lifespan ensures that we can obtain long enough motion trajectories to compute features, while also managing the processing time needed to track thousands of point in real time.

We use a window of 1 second to observe the motion trajectory and extrapolate lines for each one. We then use RANSAC to compute the vanishing point for each window. We use the coordinates of the vanishing point as a feature to train a Random Forest Classifier (max_depth = 2, 10 trees).

Phone in Hand

In the second scenario, where the user holds the phone in their hand, we collected the data in a stationary car for the safety of our participants.

Study Design Decision

We acknowledge that collecting data in a stationary car reduces the ecological validity of our evaluation. Not counting unsafe and illegal practices, there are two potential study methods with different trade-offs: (1) using a driving simulator; and (2) conducting the study in a stationary car. A task conducted in a driving simulator is able to replicate the cognitive load of actual driving, but is limited in the 'what' a camera sees while collecting the data. For each participant, it would only be able to capture the same car profile and very limited changes in lighting conditions. In contrast, an evaluation conducted in multiple stationary cars is unable to replicate the cognitive load of actual driving, but provides variance in car profiles.

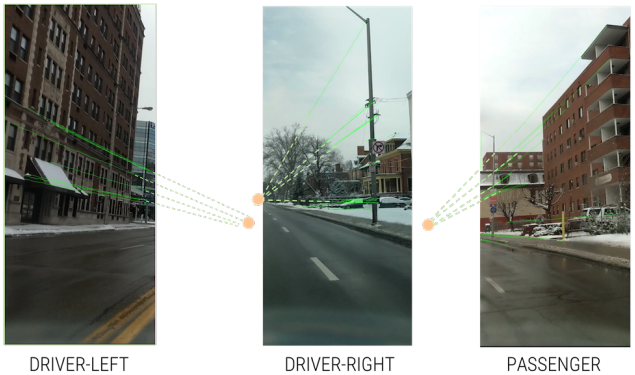

Figure 4. Image showing different vanishing point from three different positions: (a) phone on driver's left; phone on driver's right; and phone on passenger's right.

We chose to collect data and evaluate the system in stationary cars because, for a camera-based approach, variability in the scene is more critical than varying cognitive load. Moreover, there are only a limited set of positions a person could hold the phone to be able to successfully text and drive. Thus, it is more important to capture the variance in hand position. With a high number of participants, we capture variation in hand positions as well as car profiles, albeit with a trade-off against a task with higher cognitive load. Besides, we encouraged the participants to switch hand positions at random intervals and reminded them to imagine they were driving and hold the phone accordingly.

Next, we observed that a photo taken in a moving car versus a stationary car looks similar. This similarity is further confirmed by our evaluation of the docked phone in a moving vehicle. As can be seen in the video figure, our system can continuously detect faces and lines regardless of whether the car is stationary or moving. So even though the safety of the participants limits our study design, we are confident that the model developed on images from a stationary car will translate well when the vehicle is in motion.

The study was conducted over two weeks in different locations at different times of the day to ensure variability in exposure to sunlight, weather, and any influence on the image quality from environmental factors.

To elicit variance in hand positions, we did not control the phone usage behavior of the participants. They were free to use their preferred hand, and could even switch hands during the task. In fact, some participants even kept the phone in their lap or balanced it horizontally on the cup holder as they attempted to finish tasks. The repeatable patterns seen inside a car change based on *how* the person holds the phone. We compute the line-based features the same way as we did in case of docked phones. However, in this condition, the phone is typically close to a user's face and is unable to view the seat belt (as shown in Figure 5). So, we only looked for lines in the area above a person's face. Again, the number of lines detected in different frames may vary. So we make as many copies of the frame as the number of lines identified in it. We calculate the following features for each copy:

1. (x, y) coordinates of the leftmost point of the line
2. the angle of the line (in degrees) measured at the leftmost point *w.r.t.* to the left-to-right direction
3. (x, y) coordinates of the midpoint of the line

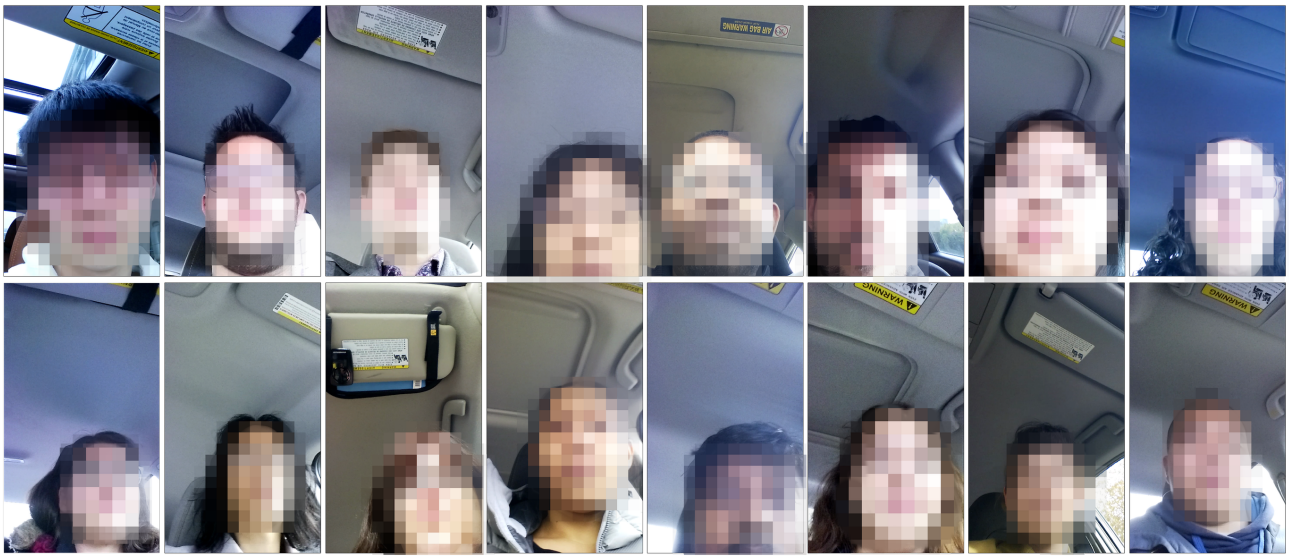

Figure 5. Examples of images captured with all 16 users where the seat belt was not visible when the phone was held.

4. (x, y) coordinates of the centre of the bounding box that encapsulates the face of the user.

We use a Random Forest Classifier (max_depth=8, 10 trees) to train our classifier with all copies of all frames as training instances. Similar to our other model, we use a majority vote among all copies of the same frame (each line). Finally, we take a majority vote of frames across a window of *m* seconds (m: 0 to 10) to distinguish between the driver and the passenger.

Figure 6. Figure showing variance in views seen from the back camera when the phone is held. (A) shows that the camera can sometimes capture car objects such as steering wheel and the infotainment system; whereas (B) shows that most times the back camera is unable to capture anything meaningful.

For the back camera, similar to docked-vent condition, the images are often occluded and do not provide a consistent signal. As shown in Figure 6-A, we can observe parts of the car such as the stereo system and determine the orientation of text (similar to our perspective of geometry approach); however, the camera may not see anything at all depending on how the phone is held as shown in Figure 6-B. Therefore, we chose only to use the front camera to build our model.

RESULTS

Note that, when not explicitly mentioned, the results are for a moving average window of size 1 second.

Docked Phone

When the front camera was used, in either *{shield, vent}*, we performed a 10 fold leave-one-car-out cross-validation. In the docked condition, we were able to distinguish between the driver and the passenger with an accuracy of 92.2% (window = 1s). We also evaluated the accuracy of our models over different window sizes. Figure 7. shows a plot of change in accuracy *w.r.t.* window size.

In our approach, we use lines as a proxy to detect the shapes of different objects such as the sun visor and the seat belt. Particularly, when we look for lines to detect seat belts, the performance of our line detection algorithm may be affected by the color and texture of clothes worn by a person. To evaluate the robustness of our approach, we conducted an additional study and recorded data (avg. length = 1min) wearing 28 different color and texture rich clothes with 2 participants (1 Male, 1 Female, mean age = 28). It includes clothes similar in color to the seat belt, and designs containing shapes that may confuse a line detection system (as shown in Figure 9 with classifier accuracy noted with each clothing). We used our best classifier from our original study, and classified this data. The average accuracy across all clothes was 91.8%. It shows that our approach is robust and can work across a wide range of colors and textures.

As stated earlier, the data from *{vent, back}* did not provide a useful signal. In the *{shield, back}* condition, we performed a 10 fold leave-one-car-out cross-validation. We were able to distinguish between the driver and passenger with 72.3%.

We observed that the majority of confusion in this case stems from confusion between the phone on passenger's right side, and the phone in the middle oriented towards the driver. For all 10 cars, there is a distinct separation between the vanishing point of the two phones. However, in 4 out of the 10 cars, the relative position was flipped as compared to the remaining cars. We postulate that this error stems from different placements of phone in different cars due to varying interior design.

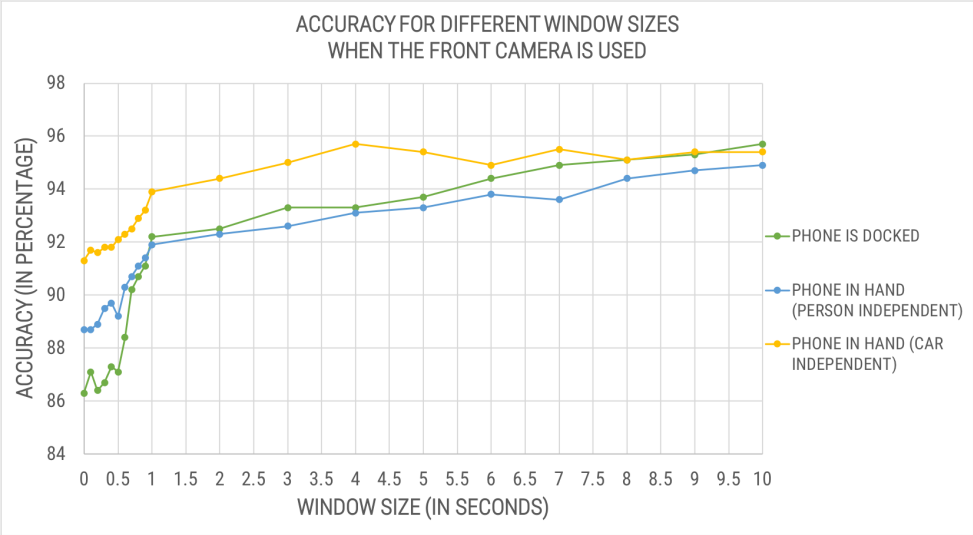

Figure 7. Plots showing accuracy of distinguishing between the driver and the passenger with varying window size when (a) phone is docked; (b) phone is in the hand (person-independent model); (c) phone is in the hand (car-independent model).

Phone in Hand

As stated earlier, when the phone is used in the hand, we only get a consistently informative signal when we use the front camera. We first validate our approach by reporting the results of a 16 fold leave-one-car-out-cross validation. We want to ensure that our machine learning model would work across different cars. In this case, we were able to distinguish between the driver and the passenger with 93.9% accuracy over a window of 1 second.

Next, we wanted to validate that our model is resilient to the variance in the way different people hold their phone while driving. We performed a 33 fold leave-one-user-out cross validation, and were able to distinguish between the driver and the passenger with 91.9% accuracy.

When a user holds the phone in their hand, the position of the phone is not static and the observed perspective might change. Figure 8 shows varying perspectives of the same car object (sun visor) in the same session. Despite this variance, we were able to distinguish between phone use by the passenger and the driver robustly. We also evaluated the accuracy of our model over different window sizes as shown in Figure 7.

Figure 8. Example of variance in image views captured at different times by the same user when the phone was held as the driver.

DISCUSSION

In this paper, we present a lightweight sensing technique to determine if the phone is being used by the driver or the passenger. Regardless of the placement of the phone in the car, we are at least able to discern between the driver and the passenger with 90% accuracy. Our continuous detection mechanism allows mobile apps to detect and adapt to the user's context *i.e.,* driving. Despite knowing that using the phone while driving is dangerous, the smartphone demands attention and causes distraction. To minimize driver distraction and improve safety, mobile apps can leverage our technique to adapt and simplify their interfaces.

We would like to emphasize that we used continuous video recording to obtain a large dataset of images. Our algorithm that uses the front camera runs on each individual frame and does not need continuous video for phone usage detection. In a real world scenario, a photo can be taken opportunistically based on event triggers such as in-vehicle detection or user touch. Secondly, our approach runs in real-time *i.e.,* the data can be featurized and processed in real time without storing any sensitive information. It allows us to detect phone usage by drivers in a privacy sensitive manner.

To validate our lightweight approach and robustness, we also built a real-time phone app. On an Octa-core 2.2 GHz Cortex-A53 android phone CPU, on average, it took 550ms to classify each frame (image).

Limitations of Our Approach

While we were able to collect data for a docked phone in a moving car, due to safety concerns, it is not possible to do so for the phone in hand condition. We acknowledge that it reduced the ecological validity of our study. But, to improve the confidence in the robustness of our system, we recruited

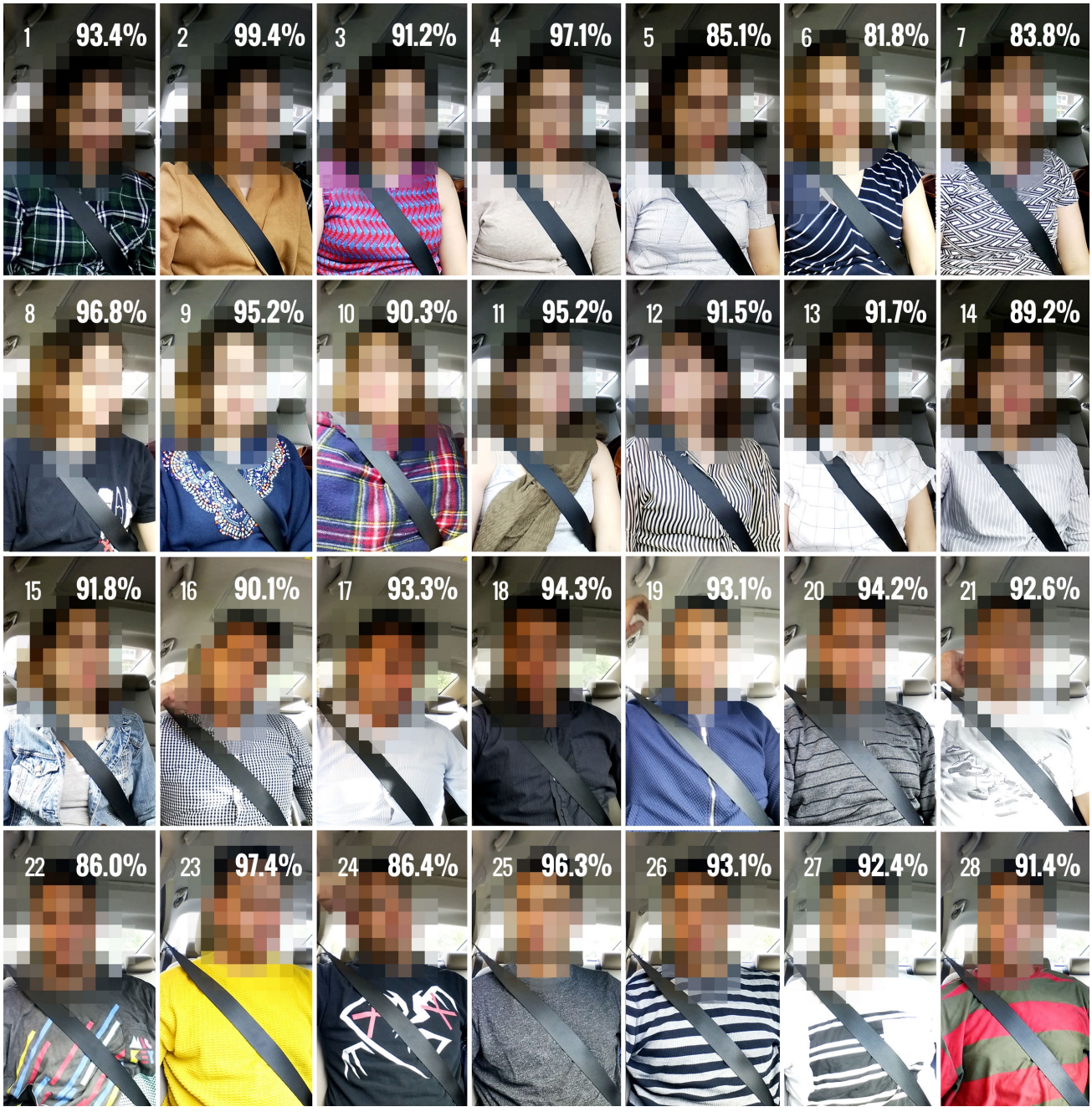

Figure 9. All the clothes used to evaluate the technical soundness of the line detection approach. Each image also shows the accuracy of data collected while wearing that cloth, using the best classifier from the original data.

Figure 10. Simplified interfaces for common mobile apps when the driver is driving.

a high number of participants. It allows us to capture a wide range of behavior exhibited by people as they pretend to drive and text. Secondly, the features used in our machine learning model do not rely on the motion of the car. We extract individual frames and only look at the geometry of the shape of the objects.

Our approach to geometry recognition using lines is not perfect. A line detection algorithm is not completely generalizable, but given our restricted search space- it works well for our use case. We have two regions of interest: (1) above a person's head to capture the geometry of the objects in a car (not affected by clothing); and (2) under the face of the person to determine the seat belt. Here, we filter out lines in a very narrow range of angles. We also conducted a study with 28 different clothing items of different colors and textures to demonstrate the technical soundness of our technique. Our results demonstrate that our approach is robust to different textures, but we cannot account for all possible textures and designs. There may be some clothing items that may cause our system to fail.

Despite an informative signal, our approach to leverage the back camera to determine the phone position/orientation did not work due to high variance in localization of the vanishing point across cars. However, the signal we obtained as quite stable, only car-specific. A potential solution could be to automatically teach the phone to build a car specific model. The phone can learn the position and vanishing point correlation, using the front camera approach as automatic position labels.

We are able to do so because our proposed front camera approach is robust and accurate. However, there are limitations to its use as well. If the phone is docked exactly in the centre of the car without any orientation towards either the driver or the passenger, then there is no observed perspective change. Our current approach may not be suitable for that scenario. But, the camera and other smartphone sensors can still be used to accomplish the same goal. The camera can observe the direction of a finger touch to determine who touched the phone screen. Prior research has also used capacitive imaging to determine the angle and direction of the finger touch on a smartphone [20], or grip information to determine hand posture to determine from which side of the car did the person touch the device [7].

If the seatbelt is occluded or camouflaged due to a person's clothes, it may adversely affect the performance of our system when the phone is docked. Our approach only uses the seatbelt as a feature when the phone is docked. So, in the absence of this signal, the aforementioned strategies of detecting finger position and angle would still work.

Lastly, if the user truly wanted to, they can fool the system. With clever positioning and unnatural orientation of the phone, a driver can trick the system into thinking that they are the passenger. This attempt to circumvent the lockout, may end up increasing the safety risk. Clearly, a complete blockage of functionality is not a good solution. So, we envision our work as a sensing platform that other apps can leverage to reduce the cognitive load of a driver and discourage bad behavior. We discuss some examples below.

Design Implications

One simple and obvious change could be to not show notifications at all when the phone is oriented towards the driver. We take it a step further to examine how commonly used mobile apps would adapt their interface to promote safety (Figure 10). We demonstrate mockups of two simplified app interfaces: (1) music application; and (2) messaging.

While the car is in motion, our system can be leveraged to detect if the phone is being used or oriented towards the driver. In such a scenario, the music app can strip away other functionalities such as music search and playlist creation. A simplified version of the music app would only present minimal options to control the currently playing song. It allows the driver to skip and change songs, but blocks the more distracting activity of music search.

Similarly, upon receiving a text message, the driver would only be allowed to use voice to text if they want to respond. While the car is in motion, the messaging app can block the driver from using text replies that demands a higher cognitive effort. The passenger on the other hand would be able to utilize the full functionality of the app. Similar feature reduction can be applied to any communication app such as messaging, Slack or even email apps. Our low fidelity prototypes are not a design guideline for future apps, rather a demonstration of how our sensing technique can be used in practice to improve safety.

However, the use of our system as a sensing platform goes beyond adapting the interface for safer driving. It can influence policies that may benefit the user. For example, if a user never drives and text, it may lead to lower car insurance rates being offered for safe driving practices.

CONCLUSION

The utility of smartphones has increased exponentially in the last decade. But, its ubiquity comes at a cost. The user often

chooses to use their phone in dangerous situations, such as while driving. Most current solutions either rely on custom hardware or are not scalable to be used in real time. In this paper, we present a fully automated, lightweight, software-only solution that leverages the on-board smartphone camera to determine if the phone is being used by the driver or the passenger. We rely on observing the change in perspective of repeatable shapes seen inside the car. We collected our data in 16 different cars with 33 different users and achieved an overall accuracy of 91.9% when the phone is held, and 92.2% when the phone is docked ($\leq$ 1 sec. resolution). A simple software update can now enable smartphones across the world to sense the context of driving and use it to adapt the mobile app interfaces to reduce distracted driving accidents.

ACKNOWLEDGEMENTS

We are grateful to the Carnegie Bosch Initiative for supporting this research.

REFERENCES

[1] National Highway Traffic Safety Administration. 2016. Distracted Driving 2014. (2016). https://crashstats.nhtsa.dot.gov/api/public/viewpublication/812260

[2] Rafael A Berri, Alexandre G Silva, Rafael S Parpinelli, Elaine Girardi, and Rangel Arthur. 2014. A pattern recognition system for detecting use of mobile phones while driving. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, Vol. 2. IEEE, 411–418.

[3] Cheng Bo, Xuesi Jian, Xiang-Yang Li, Xufei Mao, Yu Wang, and Fan Li. 2013. You're driving and texting: detecting drivers using personal smart phones by leveraging inertial sensors. In *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 199–202.

[4] Hon Chu, Vijay Raman, Jeffrey Shen, Aman Kansal, Victor Bahl, and Romit Roy Choudhury. 2014. I am a smartphone and I know my user is driving. In *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*. IEEE, 1–8.

[5] Paul Dietz and Darren Leigh. 2001. DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 219–226.

[6] Paul H Dietz, Bret Harsham, Clifton Forlines, Darren Leigh, William Yerazunis, Sam Shipman, Bent Schmidt-Nielsen, and Kathy Ryall. 2005. DT controls: adding identity to physical interfaces. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 245–252.

[7] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. 2012. GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 545–554.

[8] Zongjian He, Jiannong Cao, Xuefeng Liu, and Shaojie Tang. 2014. Who sits where? Infrastructure-free in-vehicle cooperative positioning via smartphones. *Sensors* 14, 7 (2014), 11605–11628.

[9] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 185.

[10] Hemank Lamba, Varun Bharadhwaj, Mayank Vachher, Divyansh Agarwal, Megha Arora, and Ponnurangam Kumaraguru. 2016. Me, Myself and My Killfie: Characterizing and Preventing Selfie Deaths. *arXiv preprint arXiv:1611.01911* (2016).

[11] Luyang Liu, Cagdas Karatas, Hongyu Li, Sheng Tan, Marco Gruteser, Jie Yang, Yingying Chen, and Richard P Martin. 2015. Toward detection of unsafe driving with wearables. In *Proceedings of the 2015 workshop on Wearable Systems and Applications*. ACM, 27–32.

[12] Alex Mariakakis, Vijay Srinivasan, Kiran Rachuri, and Abhishek Mukherji. 2016. Watchudrive: Differentiating drivers and passengers using smartwatches. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*. IEEE, 1–4.

[13] Mihai Negru and Sergiu Nedevschi. 2013. Image based fog detection and visibility estimation for driving assistance systems. In *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on*. IEEE, 163–168.

[14] Stanley Rabu. 2014. Detecting docking status of a portable device using motion sensor data. (March 25 2014). US Patent 8,682,399.

[15] Andrew Sears, Min Lin, Julie Jacko, and Yan Xiao. 2003. When computers fade: Pervasive computing and situationally-induced impairments and disabilities. In *HCI International*, Vol. 2. 1298–1302.

[16] Keshav Seshadri, Felix Juefei-Xu, Dipan K Pal, Marios Savvides, and Craig P Thor. 2015. Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 35–43.

[17] Chun-Che Wang, Shih-Shinh Huang, and Li-Chen Fu. 2005. Driver assistance system for lane detection and vehicle recognition with night vision. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 3530–3535.

[18] Edward Jay Wang, Jake Garrison, Eric Whitmire, Mayank Goel, and Shwetak Patel. 2017. Carpacio: Repurposing capacitive sensors to distinguish driver and passenger touches on in-vehicle screens. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 49–55.

[19] Yan Wang, Yingying Jennifer Chen, Jie Yang, Marco Gruteser, Richard P Martin, Hongbo Liu, Luyang Liu, and Cagdas Karatas. 2016. Determining driver phone use by exploiting smartphone integrated sensors. *IEEE Transactions on Mobile Computing* 15, 8 (2016), 1965–1981.

[20] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. CapCam: Enabling Rapid, Ad-Hoc, Position-Tracked Interactions Between Devices. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*. ACM, 169–178.

[21] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P Martin. 2011. Detecting driver phone use leveraging car speakers. In *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 97–108.