GazMon: Eye Gazing Enabled Driving Behavior Monitoring and Prediction

Xiaoyi Fan, *Member, IEEE*, Feng Wang, *Senior Member, IEEE*, Danyang Song, *Student Member, IEEE*, Yuhe Lu, *Student Member, IEEE*, and Jiangchuan Liu, *Fellow, IEEE*

Abstract—Automobiles have become one of the necessities of modern life, but also introduced numerous traffic accidents that threaten drivers and other road users. Most state-of-the-art safety systems are passively triggered, reacting to dangerous road conditions or driving maneuvers only after they happen and are observed, which greatly limits the last chances for collision avoidances. Timely tracking and predicting the driving maneuvers calls for a more direct interface beyond the traditional steering wheel/brake/gas pedal. In this paper, we argue that a driver's eyes are the interface, as it is the first and the essential window that gathers external information during driving. Our experiments suggest that a driver's gaze patterns appear prior to and correlate with the driving maneuvers for driving maneuver prediction. We accordingly present `GazMon`, an active driving maneuver monitoring and prediction framework for driving assistance applications. GazMon extracts the gaze information through a front-camera and analyzes the facial features, including facial landmarks, head pose, and iris centers, through a carefully constructed deep learning architecture. Both our on-road experiments and driving simulator based evaluations demonstrate the superiority of our GazMon on predicting driving maneuvers as well as other distracted behaviors. It is readily deployable using RGB cameras and allows reuse of existing smartphones towards more safely driving.

Index Terms—Gaze, driving assistant, mobile computing, deep learning

1 Introduction

Automobiles have become one of the necessities of modern life and deeply penetrated into our daily activities. They unfortunately also introduce numerous social problems, among which traffic accidents are most notoriously threatening automobile drivers and other road users. Besides well-developed passive safety equipments such as belt and air bag, active automobile safety systems are also under rapid development in recent years. They use positioning devices, built-in cameras, or laser beams to identify potentially dangerous events, so as to avoid imminent crashes. According to U.S. data [1], systems with automatic braking can reduce rear-end collisions by an average of 40 percent.

Despite being referred to as *active*, most of these systems remain passively triggered by a vehicle's surroundings and its driving interface (i.e., steering wheel, brake, and gas pedal). They react to dangerous road conditions or driving maneuvers only after they happen and are observed. Given the well-known *two-second rule*,[1] such passive reaction can greatly limit the last chances for collision avoidances. For example, the latest *Active Blind Spot Detection* system, including BMW,[2] Ford[3] and Toyota,[4] uses radar sensors to inform the driver via a symbol on the wing mirror if there is a vehicle currently in their blind spot. When the driver uses the indicator, e.g., the turn signals or the steering wheel, to change lanes, they are warned in potentially dangerous situations by a flashing LED signal and beeps, which, on a highway, can be still too late to avoid a collision if the speed is over 120 km/h. The *Adaptive Front-lighting* system, which has been developed to enhance night visibility, also follows the angle change of the steering wheel and accordingly change the lighting pattern to compensate for the curvature of a road. The lag from steering wheel movement to light movement, however, is not negligible (being activated after 1/4 turn of the wheel and sometimes one or two full turns). Another example is the navigation system, which provides rich information for a driver but can often puzzle the driver or be puzzled by the driver. When there are two close exits or intersections, frequently the driver

- *X. Fan is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China, the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada, and also with the Shenzhen Jiangxing Intelligence Inc., Shenzhen, Guangdong 518057, China. E-mail: xiaoyif@ece.ubc.ca.*
- *F. Wang is with the Department of Computer and Information Science, University of Mississippi, University, MS 38677 USA. E-mail: fwang@cs.olemiss.edu.*
- *D. Song and Y. Lu are with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. E-mail: {arthur_song, yuhel}@sfu.ca.*
- *J. Liu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China, and also with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. E-mail: csljc@ieee.org.*

Manuscript received 18 Jan. 2018; revised 17 Nov. 2019; accepted 10 Dec. 2019. Date of publication 27 Dec. 2019; date of current version 4 Mar. 2021. (Corresponding author: Jiangchuan Liu.)
Digital Object Identifier no. 10.1109/TMC.2019.2962764

1. A driver usually needs about two seconds to react to avoid accident.
2. https://www.bmw.co.uk/bmw-ownership/connecteddrive/driver-assistance/intelligent-driving#gref
3. https://owner.ford.com/support/how-tos/safety/driver-assist-technology/blind-spot/how-to-use-blis-with-cross-traffic-alert-system.html
4. https://www.toyota.ca/toyota/en/safety-innovation/safety-technology

is confused with the display, or the navigation system is confused after a wrong turn. Indeed, without a clear understanding of the driving maneuverss, the navigation system has become a major source contributing to accidents, and today's system lacks the ability to sense and predict the drivers' abnormal maneuvers beyond simply disabling touchscreen input during driving.

In short, timely tracking and predicting the driving maneuvers is essential and important towards improving driving safety, and we need a new and more direct interface beyond the traditional steering wheel/brake/gas pedal. We argue that a driver's eyes are *the* interface, as this is the first and the essential window that gathers external information. Our crowdsourcing measurements reveal strong correlations between the eye-gazing patterns and the driving maneuvers, e.g., cruising and lane change, which are further confirmed by our on-road experiments and driving simulator based evaluations to be discussed later. In particular, gaze patterns occurs prior to the corresponding driving maneuvers, which offers a great chance to overcome the two-second rule.

To this end, we develop `GazMon`, an active driving maneuver monitoring and prediction framework for driving assistance applications. GazMon extracts the gaze information from a front-camera and predicts driving maneuvers based on the gaze patterns. The patterns are analyzed through a supervised deep learning architecture. In particular, we incorporate a joint Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) network, which first identifies low-level activities, and then scales up to predict complex high-level driving maneuvers.

Our GazMon does not rely on very advanced and high-cost eye tracking devices (e.g., Tobii EyeX[5]). It is readily deployable using RGB cameras and can be easily integrated to intelligent in-vehicle systems, e.g., CarPlay and Android Auto, minimizing/reducing the reliance on extra hardware. It also allows the reuse of existing smart phones for driving maneuver prediction. It is known that smartphone usage is the second largest risk factor, resulting in 6.4 percent for crash accidents [2]. The distractions in driving mainly come from the interactions with the smartphones, such as notifications from the smartphones, watching or touching the screen. There have been solutions using mobile phones and wearable devices to capture driver maneuvers, yet few of them strive to achieve the more challenging but important task, i.e., predict driving maneuvers. Our GazMon demonstrates that a careful design can turn a smartphone from an accident contributor into a crash preventer. With GazMon, driving applications can warn and return feedbacks to drivers without distracting them, e.g., through voice instructions, to improve the safety.

We have deployed the trained deep learning models of GazMon with Mobile TensorFlow on Android smart phones, e.g., Google Pixel and Vivo X9 Plus. We conduct extensive on-road experiments for driving maneuver prediction and test distracted driving maneuver prediction on our driving simulation platform, which also provide additional feedbacks to GazMon to fine-tune the deep learning model. The evaluation results report that our GazMon not only achieves significant prediction accuracy on driving maneuvers over different state-of-art feature-based solutions, but also can successfully identify other distracted driving behaviors such as eating and reaching objects, which allows applications to benefit from predicting the driving behaviors.

The main contributions of this paper are summarized as follows:

- We present the first systematical study to thoroughly investigate the correlation between gaze pattern and driving maneuver via collecting a driving maneuver dataset that covers 129 trips in a 2-month period. The results demonstrate that gaze patterns occurs prior to the corresponding driving maneuvers (approximately 5.09 seconds on average), which offers a great chance to overcome the two-second rule.
- We propose GazMon, the first eye gazing based active driving maneuver monitoring and prediction framework, which can be implemented with the COTS smartphones, making GazMon a promising real-life deployment that can benefit many applications to actively improve driving safety.
- The GazMon implementation is the first system that jointly uses a Convolutional Neural Network and Long Short Term Memory network to effectively solve the eye gazing based driving maneuver prediction problem.
- The real-world on-road experimental results demonstrate that GazMon can allow 200 percent of the gap required by the two-second rule (i.e., 4 seconds before the actual driving maneuvers) and still distinguish various driving maneuvers and other distracted behaviors with high accuracy, which outperforms the state-of-the-art systems.

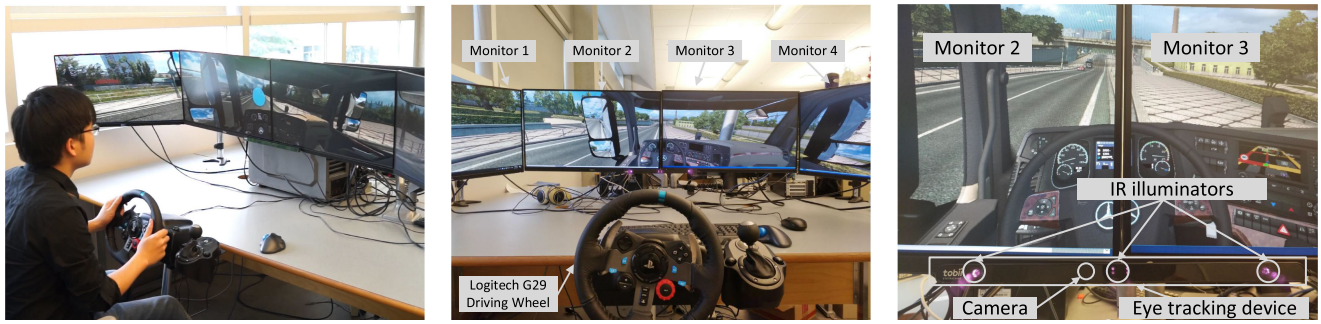
The rest of the paper is organized as follows. Section 2 presents the motivations of our GazMon in driving maneuver prediction. Section 3 provides our data preprocessing scheme to extract the features from images, and presents our deep learning approach for driving maneuver prediction. Section 4 discusses the implementation details. The performance evaluation results on our approach are presented in Section 5. Section 6 extensively discusses a series of potential applications that can be enabled/enhanced by our GazMon framework. Section 7 illustrates the related work of the research area and provides a literature review. We then conclude this paper in Section 8.

2 Motivation and Overview

2.1 Why we Incorporate Gaze Patterns Into Driving?

In this paper, we explore the opportunities to predict driving maneuvers through analyzing drivers' gaze patterns. We seek to first answer the following question: *Do a driver's gaze patterns appear prior to the driving maneuvers?* To investigate the correlations between them, we capture the driver's gaze patterns and the steering wheel through our testbed. For safety concerns, our testbed runs in a virtual reality environment as shown in Fig. 1a. The driving simulator platform runs on a customized PC, which is connected to four 27-inch monitors as shown in Fig. 1b, where the

5. https://tobiigaming.com/



(a) Driving Simulator (b) Four Monitors and Driving Wheel (c) Eye Tracking Device

Fig. 1. GazMon motivation.

NVIDIA Surround Technology enables to combine displays to create the most immersive gaming environment. As illustrated in Fig. 1c, we choose Tobii eyeX 4C[6] as the eye-tracking device to collect the users' gazing data due to its affordable price, suitable sampling rate, and reasonable accuracy. The eye-tracking device consists of three illuminators and one camera, where the illuminators create the pattern of near-infrared light on viewer's eyes, and the camera captures high-resolution images of the driver's eyes and the patterns. In this simulation platform, volunteers play a driving simulation game, namely *Euro Truck Simulator 2*, which makes people feel as driving a vehicle in real life. We record a driver's maneuvers with the gaming wheel and pedals set and capture the driver's gazing patterns with Tobii eyeX 4C to provide a straightforward comparison for our motivation.

We perform experiments over 50 experienced drivers on the gaze patterns to explore their potential relationships with the driving maneuvers. The results reveal that the driver's gaze patterns appear prior to the drivers' maneuvers, thus opening new opportunities to explore. Fig. 2 shows a typical example of the gaze patterns collected from a volunteer and the steering wheel turning maneuvers, which is the most important feature in driving a vehicle. We plot the driver's gaze patterns at the horizontal direction, where a positive degree means that the driver is looking on the left and a negative one means looking on the right. And the steering wheel turning maneuver is plotted in a similar way. It is clear to see that the gaze patterns highly correspond with the driving maneuvers but come ahead of some time advance, e.g., shoulder check comes prior to left turn for about 10-15 seconds. We count the time gap that gaze patterns appear prior to driving maneuvers, which is approximately 5.09 seconds on average and large enough for the two-second rule to apply. These observations are further confirmed by our on-road experiments and driving simulator based evaluations to be discussed in Section 5.

Then we need to answer the following question: *How a driver's gaze patterns correlate with the driver's maneuvers?* As we know, the single gaze point is ineffective to predict driving maneuvers. Our experiments reveal that if we stack the gaze points across a small time interval into a vector, then this vector can be a good indicator of different driving maneuvers. Fig. 3 shows the gaze patterns from eight typical driving maneuvers, i.e., cruising, scanning, looking at navigator, distracting, checking left side road, left turn, checking right side road and right turn. This example shows that gaze patterns are distinct with different driving intentions, and we can predict the driving maneuvers through analyzing gaze patterns.

2.2 How we Incorporate Gaze Patterns Into Driving?

Although there are a plethora of commercial off-the-shelf equipment to detect gazing patterns, most of them cannot be immediately applied to the vehicular environment. On one hand, high-end eye trackers are very expensive (e.g., Tobii Pro X3-120 costs more than $15,000), which are not suitable for wide deployment on vehicles. On the other hand, low-end eye trackers such as gaming peripherals (e.g., Tobii Eye Tracker 4C) are intended to be used in interactive and gaming applications only. Moreover, most eye tracking peripherals work with stationary computers and require calibrations with PC monitors before deployment. As such, significant effort is needed to install state-of-the-art eye-tracking devices on a vehicle, and the cost for customized modification and calibration can be prohibitively high.

Our GazeMon framework does not rely on a particular eye-tracking hardware. In the long run, advanced eye tracking solutions could be seamlessly integrated into the vehicles' onboard systems with affordable cost, and our GazeMon will benefit from it. We note that nowadays mobile phones are ubiquitous and widespread used, where more than a third of

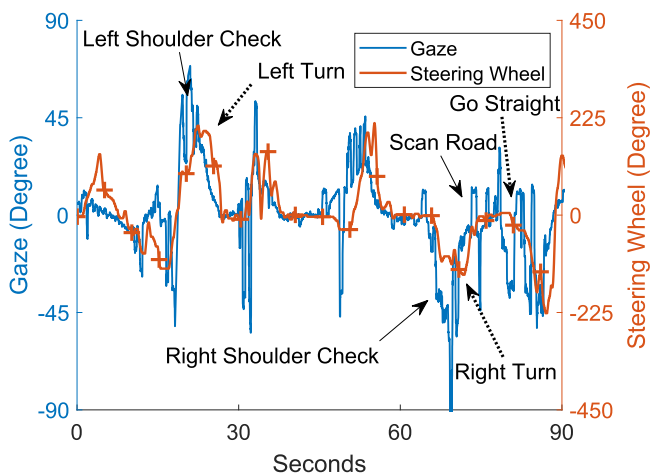


Fig. 2. Gaze patterns appear prior to driving maneuvers.

6. https://tobiigaming.com/eye-tracker-4c/

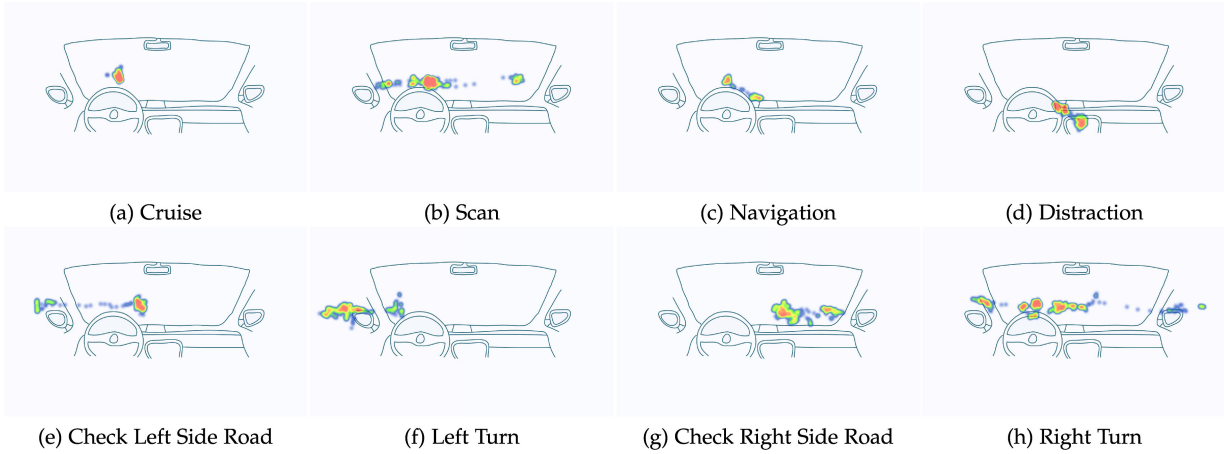


Fig. 3. Gaze patterns from typical driving maneuvers.

the world's population is estimated to have smartphones by 2019. Given that people carry their phones everyday everywhere, the phones have great potentials to serve as eye gazing tools in vehicular environments, since mobile phones can directly capture images from the front RGB camera and require no modifications to the existing on-vehicle systems. Another benefit is that the high adoption rate of technology upgrades on mobile phones can lead to rapid development and deployment of new camera technology and allow the use of computationally expensive methods. In the near future, we believe mobile phones can be a suitable and ubiquitous solution to demonstrate the importance of eye gazing and its enabled driving maneuver prediction based vehicular assistance and safety applications. Our GazMon is the first attempt towards this direction, which can achieve high prediction accuracy in a timely manner as later demonstrated by both on-road experiments and driving simulator based on evaluations in Section 5.

3 System Design

This section describes the main components of our GazMon design. Fig. 4 shows the block diagram and work flows.

3.1 Preprocessing

Our GazMon is readily deployable using RGB cameras (e.g., webcams and front-cameras on smart phones) and allows reusing existing smart phones for driving maneuver prediction. As such, GazMon can be built on top of low cost commodity-off-the-shelf (COTS) mobile phones and useful for various driving assistance and safety applications. As the image from the front-camera on mobile phones provides rich information, we propose our GazMon design to preprocess the image streaming, as illustrated in Fig. 5. We extract the facial landmarks, head pose and iris center with facial feature calibration to capture the eye gaze information for the driving maneuver prediction.

3.1.1 Facial Landmarks Detection

We follow the notation in [3] for facial landmark detection. Let $\mathbf{x}_i \in \mathbb{R}^2$ be the x, y-coordinates of the ith facial landmark in an image I. Then the vector $\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\}$ denotes the coordinates of all the p facial landmarks in I. In this paper, we refer to the vector $\mathbf{S}$ as the shape, and use $\mathbf{S}(t)$ to denote our facial landmark estimation on the image I_t at time t. Each time t regressor $r_t(\cdot, \cdot)$ in the cascade predicts an update vector from the image I_t and the current shape estimation $\mathbf{S}(t)$, and then is added to the current shape estimation $\mathbf{S}(t)$ to calculate $\mathbf{S}(t+1)$

$$\mathbf{S}(t+1) = \mathbf{S}(t) + r_t(I_t, \mathbf{S}(t)). \quad (1)$$

The critical point of the cascade is that the regressor r_t makes its predictions based on features (e.g., pixel intensity

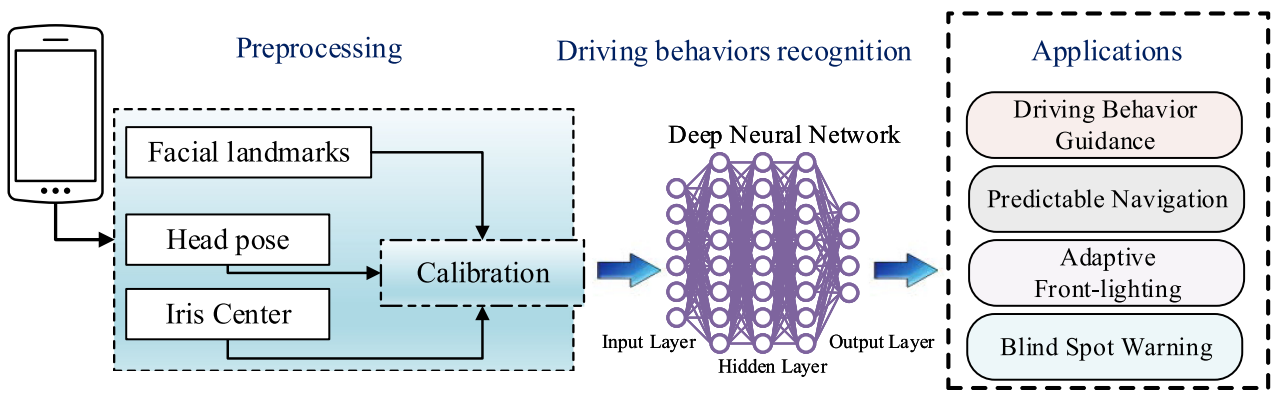


Fig. 4. GazMon framework.

(a) (b) (c) (d)

Fig. 5. A streaming of images for right shoulder check.

values) computed from I_t and indexed relative to the current shape estimation $\mathbf{S}(t)$.

3.1.2 Head Pose Estimation

In real driving scenarios, drivers change their head pose and facial expression, while their head position keeps relative stable. The accuracy of capturing eye gaze information highly depends on the head pose estimation. Accurately estimating the driver's head pose in complex situations is a challenging problem. GazMon extracts head pose rotation information in addition to facial landmark detection. We start with single camera tracking by exploring the 2D-3D spatial consistency of feature points on each image. Given the tracked 2D facial landmarks, we use the POSIT algorithm [4] to estimate user's head pose. The POSIT algorithm finds an object's pose by iteratively minimizing the error between the projection of a known 3D model and the 2D landmarks tracked. When the 2D facial features are tracked in each image, the 2D to 3D conversion method can be utilized to obtain the head pose information. Given an input image I_t and facial landmarks $\mathbf{S}(t)$, $\mathbf{P}(t)$ is estimated by minizing

$$e(t) = \sum_{k=1}^{K} \rho(\mathbf{s}_k(t) - \phi(\mathbf{P}(t)\mathbf{S}(t))), \tag{2}$$

where $\phi(\cdot)$ is the perspective projection of a 3D point from the model in homogeneous coordinate $\mathbf{S}(t)$ to its 2D correspondence $\mathbf{s}_k(t)$ and $\rho(\cdot)$ is an M-estimator chosen to alleviate gross noise interference. The POSIT algorithm is efficient with acceptable accuracy, and thus adequate to real-time applications.

3.1.3 Iris Center Detection

Our real world experiments have shown that the visible imaging methods are not robust to the lighting conditions. For example, pupils may not be visible without IR lighting, which is the key stone for eye tracking. We instead use the iris center as a notable feature, by which we can estimate the gaze direction. Our iris detection is based on [5]. The eye region $\mathbf{E}(t)$ is extracted from the facial landmarks, and the iris center is then detected in the eye region. To achieve this, we first use the L_0 gradient minimization method [6] to smooth the eye region, where a rough estimation of the iris center can be obtained by the color intensity. Random sample consensus (RANSAC) is then utilized to estimate the radius r of the iris. At last, a combination of intensity energy and edge strength information is utilized to locate the iris center, where the intensity energy in the circle window should be minimized and the edge strength of iris edges should be maximized.

3.1.4 Eye Gazing Feature Calibration

To work ubiquitously, the system should be able to calibrate itself to different mobile phone placements and vehicle models. Here the main challenge is that it is hard for a user to place the mobile phone exactly on the same position of the dashboard every time. We thus develop a calibration module to automatically align the smart phones to a fixed coordinate system. The calibration is done by collecting an initial measurement that takes about 10 seconds for the driver to keep looking at the front. In particular, we construct the feature frame $\mathbf{F}(t)$ that denotes the measured eye gaze information at time t and $\mathbf{F}(t) = \{\mathbf{S}(t), \mathbf{P}(t), \mathbf{E}(t)\}$. Let $\tilde{\mathbf{F}}$ represent the eye gazing information of the 10 second initial measurement. We map the measured eye gazing information $\mathbf{F}(t)$ at time t to the calibrated eye gazing information $\hat{\mathbf{F}}(t)$ as follows:

$$\hat{\mathbf{F}}(t) = \mathbf{F}(t) - \tilde{\mathbf{F}}. \tag{3}$$

3.2 Deep Learning Architecture for Driving Maneuver Prediction

This part starts from the design of our feature frame. The preprocessing stage outputs the eye gazing features for each image, which we utilize to build the frame. Specifically, we provide the following as input to the model: (1) the facial landmarks together with their locations in the image, (2) the head pose, and (3) the iris centers. The size of facial landmark frame is $n \times L$, where n is the sampling rate and L is the number of landmarks in one image. The size of head pose frame is $n \times 3$, which corresponds to the 3D information. And the size of eye irises is $n \times 4$, where there are two irises on each face and the coordinate of each iris has two values in the form of (x, y). As illustrated in Fig. 6, we construct a deep learning architecture that jointly uses a Convolutional Neural Network (CNN) [7] and a Long Short Term Memory (LSTM) network [8]. In our deep learning architecture, the CNN is used to extract spatial relationships in a single frame, and LSTM is used to learn dynamic temporal relationships from a sequence of frames. Our deep learning design takes the results from the data preprocessing as inputs into the neural networks. CNN outputs are then processed through two layers of stacked LSTMs. The output is the classification of maneuvers using a softmax layer. We discuss the layers one by one in the remainder of this subsection.

We use the fully-connected layer to merge the three inputs, where these features are outputs of rectified linear units. As

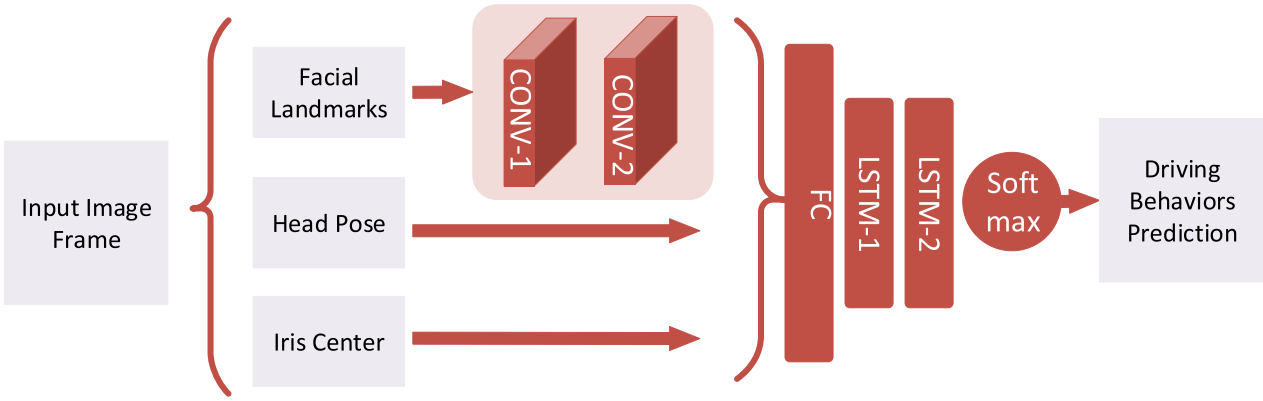


Fig. 6. GazMon deep learning architecture.

illustrated in Fig. 6, CONV represents convolutional layers (with filter size of kernels: CONV-1: $n \times 16$, CONV-2: $n \times 4$), while FC represents the fully-connected layers and LSTM represents LSTM layers. We construct a CNN to take the facial landmarks frames as input and provide the output to be fed into the fully-connected layer with head pose and iris center information. The merged features then form the input as a sequence of frames to the LSTM structure.

LSTM is a structure originally proposed by Hochreiter and Schmidhuber [8]. An LSTM cell is a subnet that allows to easily memorize the context information for long periods of time in sequential data. The subnet includes three gates: the input gate i_t, the forget gate f_t, and the output gate o_t, which have the controls to overwrite, keep, or retrieve the memory cell c_t, respectively. Each LSTM cell remembers a single floating point value c_t. This value may be diminished or erased through a multiplicative interaction with the forget gate f_t or additively modified by the current input x_t multiplied by the activation of the input gate i_t. The output gate o_t controls the emission of the memory value from the LSTM cell.

Let $\sigma(x) = (1 + e^{-x})^{-1}$ be the sigmoid function, which controls the inputs to a range of [0,1]. We have

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (7)$$

$$h_t = o_t \tanh(c_t), \quad (8)$$

where the W terms denote weight matrices (e.g., W_{hi} is the input-hidden weight matrix), and the b terms denote bias vectors (e.g., b_f is the bias vector of forget gate).

The LSTM cells are then grouped and organized into a deep LSTM architecture. Inside the architecture, the output from one LSTM layer will be the input for the next LSTM layer. We fine-tune the LSTM architecture with various numbers of layers and memory cells, and choose to use two stacked LSTM layers, each with 32 memory cells.

Following the LSTM layers, a softmax classifier is used to make a prediction at every image I_t, i.e., to get the probability distribution over the maneuver label y in the maneuver cluster γ

$$Pr(y|I_t) = \frac{e^{I_t}}{\sum_{y' \in \gamma} e^{I_t}}. \quad (9)$$

Our goal is to find the maximum likelihood of all training samples. As an objective function, we apply the negative log probability, i.e., the cross entropy error function

$$E = -\sum_{y \in \gamma} z_y ln Pr(y|I_t), \quad (10)$$

where $z_y \in \{0, 1\}$. $Pr(y|I_t)$ is the predicted probability of the class with maneuver label y.

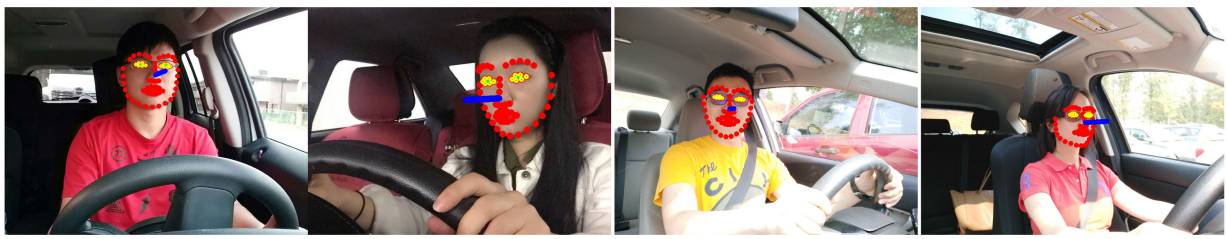
4 System Implementations

We have implemented GazMon on various mobile device hardware, such as Google Pixel, Xiaomi Mi Max, Vivo X9 Plus, Huawei Nexus 6P and LG Nexus 5. Fig. 7a shows our GazMon running on implemented Vivo X9s Android smartphone with a 20-Megapixel front camera, which is placed on the vehicle's dashboard. During the 10 second measurement of the calibration, the driver needs to keep looking at the front to allow the capturing of an initial pose of the user's head and enough facial information. In this stage, the driver holds their head still while looking ahead. The video from the front camera is captured and used to build an initial head model, which requires facial landmarks, iris center and head pose have stable values. When checking the left blind spot, even the camera sometimes can only capture half of the driver's face, the missing information can be partially recovered based on euclidean distance among facial landmarks in the initial head model, which will then be further processed by our deep learning model that is capable of using dynamic temporal relationships from a sequence of frames (note that not all the frames only capture half of the driver's face) to better predict the driver's maneuvers.

The mobile phone part of GazMon is implemented as an app on Android OS 5.1.1 working together with dlib-android API[7] and TensorFlow Mobile Android API,[8] which is based on JAVA in the android-studio programming environment. On startup, the GazMon app launches an Android activity (CameraActivity.java) which basically accesses the camera by using the Android Camera2 package. Then

7. https://github.com/tzutalin/dlib-android
8. https://www.tensorflow.org/mobile/

(a) Smartphone Position (b) Vivo X9 Plus on 2015 Ford Mustang (c) Google Pixel on 2014 Audi S5 (d) Nexus 6P on 2016 Mazda CX-5



(e) Xiaomi Mi Max on 2015 Jeep Patriot (f) Vivo X9 Plus on 2016 Toyota Camry (g) LG Nexus 5 on 2013 Honda Civic (h) LG Nexus 5 on 2016 Mazda CX-5

Fig. 7. (a) shows the real-world experiment, and (b)-(h) show example snapshots of the dataset.

GazMon uses the supported Java Native Interface (JNI) procedures to interact with dlib-android engine and the recent proposed dlib library [9] to extract a sequence of eye gazing features from the incoming image stream.

In training stage, GazMon uploads the drivers' videos with the preprocessed eye gazing features in a batch to the server, when the high-speed wireless connection is available. The preprocessed eye gazing features are used for training the deep learning architecture, where the ground truth of the driving maneuvers is labeled based on the videos from the front cameras. Figs. 7b, 7c, 7d, 7e, 7f, 7g, and 7h show the video snapshot examples and the eye gazing features in our dataset. The server part of the GazMon is deployed on our customized desktop, where CNN and LSTM classifiers are implemented in Keras[9] with cuDNN on Dual Nvidia GTX 1080Ti GPUs.

In prediction stage, the GazMon app running on smart phones can timely process the images captured by the device's camera and predict the driving maneuvers based on the deep learning architecture pre-trained by the aforementioned approach, so as to provide realtime services to users, where the preprocessed eye gazing features are fed into TensorFlow Mobile's core engine implemented by Google developers.

5 On-Road Test and Further Investigation on Distracted Driving Behavior

It is well known that in the auto manufacture industry, developing a new technology on vehicles, especially driving safety system, involves an iterative process, including many rounds of design, validation and improvement. GazMon is no exception. To the end, we conduct extensive real-world experiments with GazMon to evaluate and further improve its performance. The experiments include six vehicle models (i.e., 2015 Jeep Patriot, 2016 Mazda CX-5, 2015 Ford Mustang, 2016 Toyota Camry, 2016 Lexus IS and 2014 Audi S5) and 10 drivers driving on two typical types of roads (i.e., highway and urban streets), covering 129 trips with 2,469 minutes in a 2-month period. To evaluate the prediction quality, 6,120 driving maneuver examples are labeled based on the real-world videos. To conduct a comprehensive on-road study, we test five typical driving maneuvers, i.e., cruising (CR) with 1,796 examples, left turn (LT) with 1307 examples, right turn (RT) with 1,449 examples, left lane change (LL) with 728 examples and right lane change (RL) with 840 examples, as shown in Figs. 8 and 9. Due to the safety concerns, the distracted driving behaviors (e.g., using cell phone, reaching for a moving object, eating or drinking) cannot be involved in on-road study, which are tested on our simulation platform and presented at the end of this section.

We train the models for the two different scenarios with cross validation to mitigate overfitting, where 80 percent of the data is used as a training set and the remaining 20 percent is used as a test set. The training includes 100 epochs using stochastic gradient descent (SGD). We implement the CNN networks with two convolutional layers with a dropout of 0.5, followed by one fully-connected layer to merge head pose and iris center information. Our LSTM networks use 32 memory cells per layer. Throughout training, we save the model

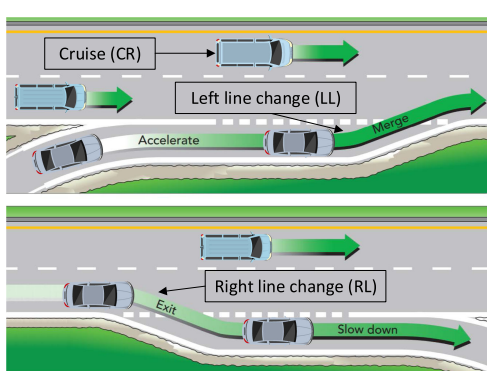


Fig. 8. Examples of cruise, left lane change and right lane change [10].

9. Keras: Deep Learning library for Theano and TensorFlow. https://keras.io/

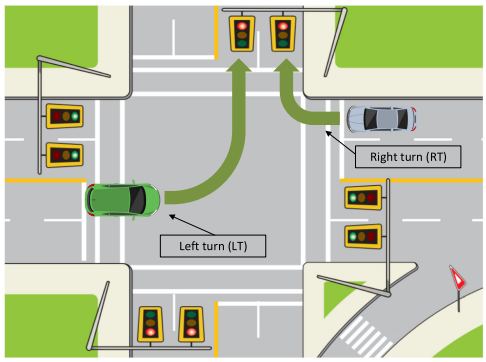


Fig. 9. Examples of left turn and right turn [10].

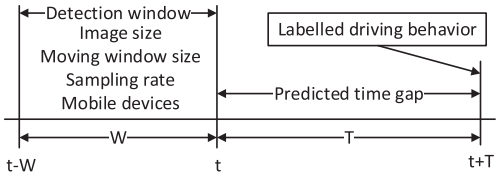


Fig. 10. Experimental parameters.

and compute prediction accuracy on the test set for each epoch. As depicted in Fig. 10, at time t, we use the eye gazing information collected during time $[t-W, t]$ to perform the prediction for the maneuver at time $t+T$, where W is the detection window size and T is the predicted time gap. We use different experiment parameters to allow investigating different predicted time gap and various aspects of detection window, e.g., image size, detection windows size, sampling rate, and different mobile devices.

5.1 On-Road Experiment Results

Table 1 shows the details of prediction accuracy in precision (P), recall (R) and F-Score (F) [11] of our *GazMon* approach, where each column denotes the driving activity performed and each row represents the prediction time gap. As shown in the table, the precision of driving maneuver prediction is at least 0.96 in 4 seconds, which indicates that GazMon can allow 200 percent of the gap required by the two-second rule and still distinguish various driving maneuvers with high accuracy. We thus use 4 seconds as the default predicted time gap for the remained experiments. We also observe that the left lane change (LL) has better prediction accuracy than the right lane change (RL) in longer predicted time gap, because the left lane change takes longer time as the vehicle needs accelerate to merge into the left

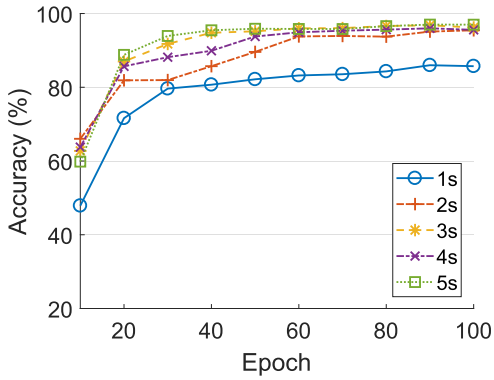


Fig. 11. Moving window size.

lane. When the predicted time gap is larger than 5 seconds, the prediction accuracy decrease for the right lane change (RL) and left turn (LT). This is because the experienced drivers always have right shoulder check before both of those maneuvers. If the predicted time gap is too large, it will cause that the prediction is mainly based on the right shoulder check and thus cannot well distinguish these two maneuvers.

We next examine the accuracy of driving maneuver prediction when applying different detection window size. Fig. 11 illustrates that the larger the detection window size is, the greater the accuracy achieves. Meanwhile, too large a detection windows size can lead to huge computation resource consumptions on mobile phones, which may cause excessive delay for the maneuver detection. We observe that with a 3 s detection window size, our system achieves more than 90 percent prediction accuracy, which indicates that our system can achieve high prediction accuracy under a small computation latency. We thus use this value as the default detection window size. We also evaluate the prediction accuracy with different sampling rates. In particular, Fig. 12 illustrates that the accuracy drops for low sampling rate, and the higher sampling rate makes the accuracy better. Similar to the detection window size, too higher sampling rate will exhaust the mobile phones' computation resources and thus slow down the maneuvers prediction. To this end, we use a sampling rate of 10 Hz as the default setting of our evaluation, which yields an average accuracy of 94 percent.

Fig. 13 shows the performance of our GazMon compared with different state-of-the-art approaches. To this end, we implement five commonly used classifiers (k-Nearest Neighbors, one-versus-all Linear SVM, Decision Tree, Random Forest, and Quadratic Discriminant Analysis) as well as the CNN

TABLE 1
The Accuracy of Driving Maneuver Prediction Versus Prediction Gap

	CR			LT			RT			LL			RL		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
1	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
2	1.00	1.00	1.00	0.98	1.00	0.99	1.00	0.98	0.99	0.96	1.00	0.98	1.00	0.97	0.99
3	0.97	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.99	0.99	0.97	0.98
4	0.96	1.00	0.98	1.00	1.00	1.00	1.00	0.95	0.98	0.98	0.98	0.98	0.99	0.99	0.99
5	0.96	0.87	0.91	0.94	0.92	0.93	0.88	0.94	0.91	0.88	0.98	0.93	0.95	0.90	0.92
6	0.96	0.98	0.97	0.84	0.96	0.90	0.89	0.91	0.90	0.84	0.91	0.88	0.97	0.78	0.86
7	0.94	0.93	0.94	0.93	0.61	0.74	0.98	0.94	0.96	0.94	0.98	0.96	0.75	0.96	0.84
8	0.98	0.98	0.98	0.85	0.57	0.68	0.97	0.90	0.93	0.83	0.96	0.89	0.72	0.87	0.79

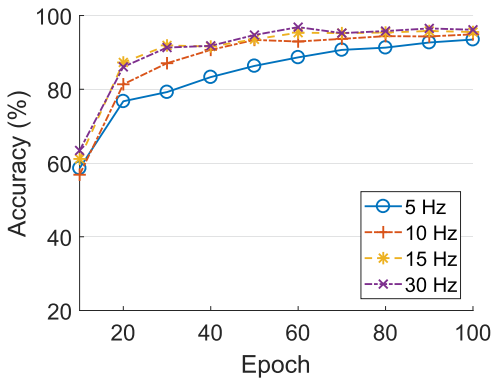


Fig. 12. Sampling rate.

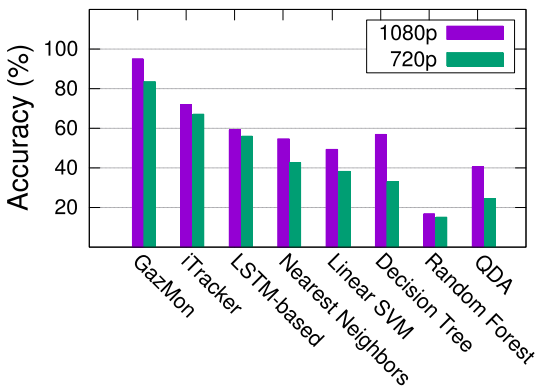


Fig. 13. Overall performance of GazMon system.

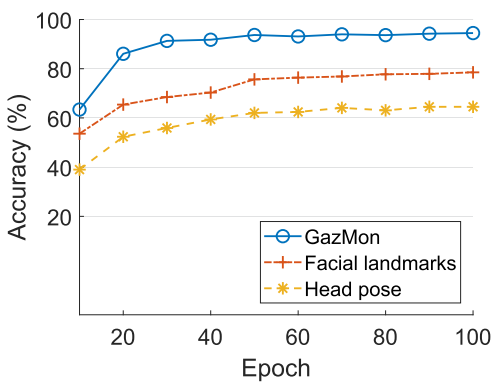


Fig. 14. Different inputs.

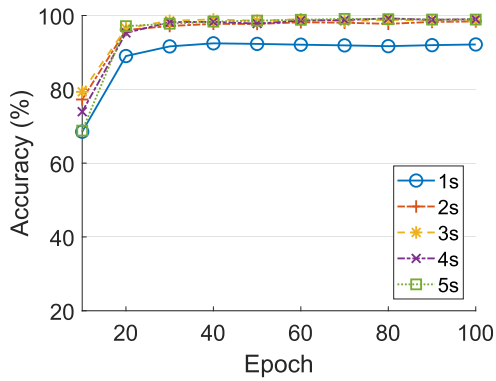


Fig. 15. Highway.

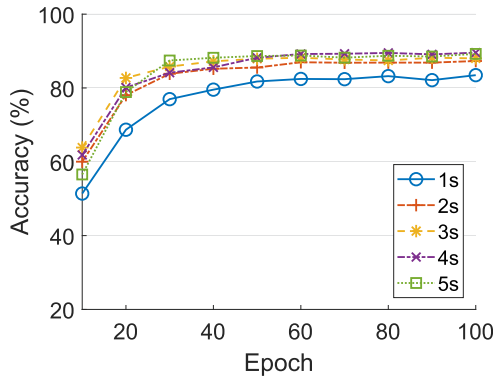


Fig. 16. Urban street.

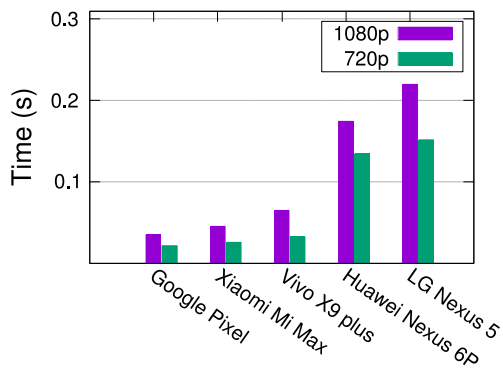


Fig. 17. Time cost.

based approach used in iTracker [12] and a LSTM based approach. The result clearly shows that GazMon can achieve 22 percent higher accuracy than iTracker that only uses CNN. This demonstrates the benefits of the LSTM architecture used in GazMon on learning dynamic temporal relationships from a sequential spectrum frames for driving maneuver prediction. At the same time, GazMon also obtains 36 percent higher accuracy than the LSTM-based approach, which illustrates the necessity of the CNN architecture used in GazMon to efficiently extract the features for driving maneuver prediction. Our GazMon also outperforms the other five commonly used classifiers, achieving 40 percent gain over the best approach (SVM) among them. One general observation is that as the 1080p images contain more details for facial features, especially the eye areas, which provide more opportunities to achieve higher prediction accuracy.

In Fig. 14, we compare the results of our deep learning design with inputs from various preprocessing options. The comparison among the facial landmarks, head pose, images and GazMon shows the effectiveness of our preprocessing scheme for obtaining eye gaze information, which potentially offers the rich information for driving maneuver prediction and leads to a better performance against head-based driver monitoring systems.

Figs. 15 and 16 show the driving maneuver prediction accuracy in two different environments, i.e., along with the highway and in the urban streets, respectively. Our GazMon achieves the best performance along the highway environment with the accuracy of up to 99 percent, and the prediction accuracy in the urban streets is up to 89 percent. The difference is mainly because the diversity of driving maneuvers along the highway is much lower than in the urban streets. We further examine the performance of GazMon with different mobile devices, as shown in Figs. 17 and 18. The mobile

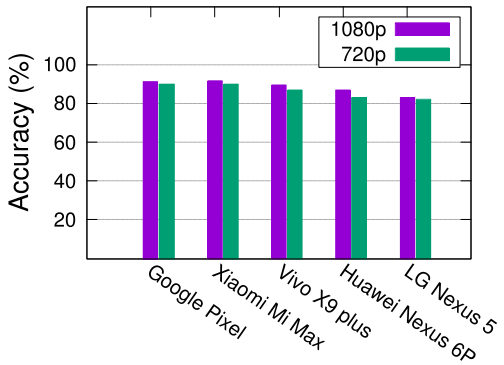


Fig. 18. The accuracy on different mobile devices.

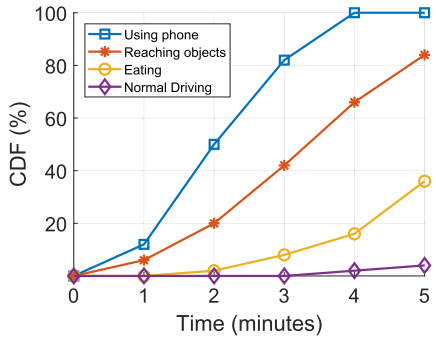


Fig. 19. Driving time until accidents occur.

TABLE 2
The Accuracy of Normal and Distracted Driving Behavior Prediction

	Normal Driving			Distracted Driving		
	Precision	Recall	F-score	Precision	Recall	F-score
1	1.00	0.99	0.99	0.98	1.00	0.99
2	0.99	0.99	0.99	0.98	0.98	0.98
3	0.99	0.98	0.98	0.96	0.98	0.97
4	1.00	0.96	0.98	0.95	1.00	0.98
5	0.99	0.94	0.96	0.91	0.98	0.95
6	0.99	0.89	0.94	0.85	0.98	0.91
7	0.89	0.97	0.93	0.96	0.85	0.90
8	0.88	0.92	0.90	0.89	0.83	0.86

devices include Google Pixel, Xiaomi Mi Max, Vivo X9 plus, Huawei Nexus 6P and LG Nexus 5, in the order of their processor performance. Fig. 17 shows the processing time for a single frame image on different devices. Intuitively, the larger image takes longer time to process. Taking Google Pixel as an example, which is released on October 2016, yet the processing time is much less than 0.05 s, meaning that GazMon on it can easily process the 1080p image streaming with the rate of 10 frames per second. Even for LG Nexus 5, which is released on October 2013 and has the slowest processor among these mobile devices, can still process a 720p image within 0.15 second (i.e., a 720p image streaming of at least 5 frame per second), indicating that GazMon can work on a variety of mobile device hardware. In Fig. 18, GazMon can achieve a high and relatively stable prediction performance. Even for LG Nexus 5, the least powerful hardware that can only process about 5 frames in one second, the average accuracy is still well above 80 percent.

5.2 Further Investigation on Distracted Driving Behaviors

As on-road experiments with distracted driving behaviors, e.g., eating and reaching objects, can be dangerous and may cause safety and ethical issues, we therefore evaluate the performance under such behaviors on our driving simulator instead. In the risky distracted driving simulation, we invite 50 experienced drivers as volunteers who are varied in age and gender. In each simulation case, the volunteer plays the first mission[10] in the driving simulation game, namely *Euro Truck Simulator 2*, which starts from Frankfurt to Mannheim with 82 km distance and takes around 10 minutes to drive in the game. To conduct a comprehensive evaluation, drivers will drive as normal first and then engage in three typical risky distracted driving behaviors, i.e., (1) using cell phone, including texting, dialing and reading; (2) reaching for a moving object; and (3) eating snacks or drinking nonalcoholic beverage.

We first report the precision, recall and F-Score in Table 2 for distracted driving behaviors. GazMon can achieve the performance of 0.95 (precision), 1.00 (recall), 0.98 (F-Score), when the prediction time gap is 4 second. This again demonstrates the effectiveness of our GazMon on driving behavior prediction, especially considering the two-second rule for accident avoidance. Then we evaluate the relationship between distracted driving behaviors and accidents, including speeding or red light tickets, car/barrier crashes and driving on the wrong lanes. We plot of the empirical CDF of driving time until accidents occur in Fig. 19, which gives a much clearer picture of the importance of predicting the driving behaviors. We can see that the distracted driving behaviors have a remarkable impact on the accidents. Taking using phone in driving as an example, nearly 80 percent of the drivers have accidents in 3 minutes of the simulated driving. On the normal driving case, we require the volunteer carefully paly the driving game, the driving accidents can be reduced to less than 10 percents, indicating that avoiding distracted driving behaviors can effectively reduce accidents and keep the drivers safe. We further explore accidents distribution versus distracted driving behaviors in Fig. 20. In the using phone case, 70 percent drivers will have

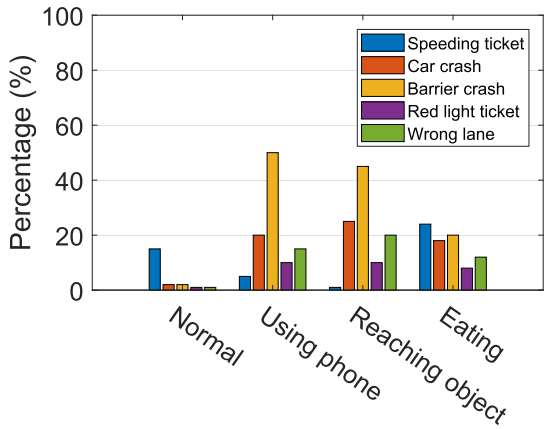
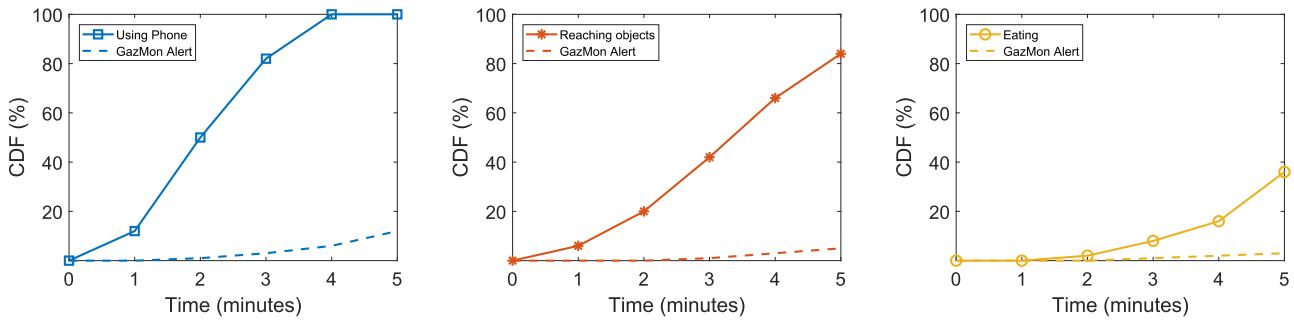


Fig. 20. Accidents distribution versus distracted driving behaviors.

10. https://www.youtube.com/watch?v=Z1FAOuylvzg



(a) Using phone (b) Reaching objects (c) Eating

Fig. 21. Drivers will stop their distracted driving behaviors after GazMon alerts.

car or barrier crashes. In the normal driving case, we can see that most accidents are speeding tickets, since it is hard to provide drivers the full feeling of speed on the simulator. To better evaluate the potentials of GazMon for driving safety, we conduct an experiment where drivers will have distracted driving behaviors once per 30 seconds, and we let GazMon alerts the driver as soon as a distracted driving behavior is detected. When GazMon triggers the alerts, drivers should stop their distracted driving behaviors and pay their attention to driving. The results are shown in Fig. 21, which illustrates the accident rates are significantly reduced, indicating a significant improvement to the driving safety.

6 Discussions on Potential Applications

Our performance evaluation has demonstrated the effectiveness of GazMon, which with a simple alert mechanism, e.g., beep sound alerts, can already greatly help reduce accidents and improve driving safety. Indeed, we envision that our GazMon can further facilitate various safety and driving applications. We discuss some of them in the following subsections.

6.1 Driving Behavior Guidance

One of the critical applications for GazMon is the distractions detection. Driver distractions have been identified as a common causal factor in vehicle collisions by extensive researches. The drivers' gazing patterns can provide useful information for addressing this issue. For example, the gaze pattern based on online maps for a specific road can be utilized to remind drivers via predicting driving styles and warning dangerous driving behaviors.

Another possible enhancement by our GazMon might be to use crowdsourcing to determine the zones that the drivers should pay attention to. Those zones can be figured out by using historical drivers' gazing patterns. For example, by comparing estimated and expected gaze patterns at a specific location, the system can warn the driver visually if the drivers overlook those zones.

6.2 Predictable Navigation

With the advances of outdoor positioning services (GPS in particular), the navigation for drivers or pedestrians has increasingly been an essential application on smartphones or car consoles. Real-time driving information, such as live traffic or construction locations, has been incorporated as well. The quality of the recommended routes is generally acceptable with state-of-the-art navigation services; the interaction techniques however are far from convenience, although voices commands [13] and gesture control have been integrated into in-vehicle interfaces.

It is known that the state-of-the-art navigation systems do not have the ability to sense the users' status or intentions. In fact, quite often a driver may get puzzled during the driving, e.g., searching for the exits along the highway, or confused with the driving lanes. Google Maps and similar navigation services provide less detailed guidelines or simply fail, since the interfaces of navigation systems limit the interactions with the driver. The driver is difficult to ask the navigation system for more routes details, not to mention to let navigation systems sense the driver's determinedness. Yet, with our GazMon and the augmented reality technology, the navigation system can detect the driver's uncertain status and provide more detailed guidance to drivers.

6.3 Other Applications

Our GazMon can potentially lead to many other interesting applications. One of them would be the adaptive front-light system. The state-of-the-art systems mainly depend on vehicle speed and steering input. With the information from our GazMon, the driver could naturally interact with the vehicle by his/her gaze moving, where the adaptive front-light can point the low-beams headlights to the direction the driver intends to observe, so as to enable the front-light system to work based on the drivers' observation intentions rather than the steering wheel turning behaviors.

Another application that enhances driver safety is the blind spot warning systems. For automotive driving, blind spots are zones outside of a vehicle that the driver is unable to see. The blind spot warning systems detect other vehicles located to the driver's side and rear. For example, the system can provide the driver with a warning, if a vehicle enters the driver's blind spot while s/he is changing lanes. Yet such blind spot zones are pre-defined, and will not alert drivers to pedestrians, bicyclists, animals or vehicles outside its detection zones. There exist however many accident risks in the areas that the driver may overlook, where threat levels can be further assessed based on our GazMon with timely warnings being provided if necessary. Another example application can be to detect and alert driver's bad driving habits with the help of our GazMon, which may need some extension to integrate with other information such as steering wheel movement. When the driver

actually starts turning the steering wheel, the system could determine whether the driver checks the mirrors, glances over the shoulder to check blind spot areas and uses the turn signal. If not, the system could warn the bad driving behaviors, no matter there will be impending risks or not.

7 Related Work

There has been a plethora of works on eye detection and tracking [14], which can be classified into two categories, *infrared imaging* and *visible imaging*, depending on whether they require external light sources (infrared lights) or not. The infrared imaging method, e.g., Tobii Eye Tracker 4C,[11] needs infrared cameras and infrared light sources, where the latter can be used for controlling light conditions, obtaining higher contrast images and for stabilizing gaze estimation. This makes infrared imaging method able to reduce the effects of light conditions, and produce a sharp contrast between the iris and pupil. However, the infrared imaging method requires multiple light sources to improve the usability of the eye gaze tracking technology. In contrast to that, there are a number of works on eye tracking algorithms using an RGB camera, which are known as *visible imaging* method. The visible imaging methods do not need special cameras and light sources, and thus are more widely utilized, which are further grouped into two categories, i.e., appearance-based and iris-based gaze tracking approaches. Appearance-based methods attempt to build a mapping from the appearance of the eye to the gaze point on screen coordinates. Along this line, Sugano *et al.* [15] presented an online learning algorithm within the incremental learning framework for gaze estimation, which utilized the users operations (i.e., mouse click) on the PC monitor. At each mouse click, they created a training sample with the mouse screen coordinate as the gaze label associated with the head pose and eye image. Later, to reduce the training cost, Lu *et al.* [16] introduced an adaptive linear regression model to infer the gaze from eye appearance. However, as the appearance may be inconsistent due to factors such as illumination changes and head movement, appearance based approaches [15], [16] rely on significant amount of training data and require extensive computation resources, which are not suitable on mobile devices. Iris-based gaze tracking relies on extracting the features of the eye region, e.g., the iris center and iris contour, to provide eye movement information. Wang *et al.* [17] first detected iris through an ellipse fitting procedure. The shape of the ellipse can be used for determining the normal of 3D iris. Valenti *et al.* [18] infered gaze directions from observed eye shapes, such as pupil center or iris edges. Yet, Iris-based approaches [17], [18] are not accurate, due to extracting the exact shape of iris is often very difficult. Recently, Krafka *et al.* [12] proposed to collect the face images as inputs, and apply a deep learning approach for gaze tracking. Different from all these aforementioned approaches, we carefully construct a novel deep learning architecture to coherently utilize pre-processed eye gazing information, such as facial landmark, head pose and eye center location, so as to achieve a high accuracy solution for driving behavior prediction.

On the other hand, mobile applications have been witnessed an explosion on vehicles [19], [20] to reinforce driving safety. Such pioneer applications [19], [20], [21] to detect whether the mobile device user is a driver or passenger, which can facilitate many applications aiming to eliminate distracted driving. Sodhi *et al.* [22] demonstrated eye movements can be collected and analyzed to compare a driver's performance with head-mounted eye-tracking devices to track on-road driver eye movement, including one computer and three different COTS cameras, i.e., a scene camera, an eye camera and an IR camera. Qi *et al.* presented DrivAid [23] to infer different driver maneuvers based on drivers head pose changes. The system leveraged audio-visual to augment driving behavior analysis based on IMU sensors and inferred the focus of attention by tracking the head orientation. Doshi *et al.* [24] presented a study for the eye gaze and head movement to predict driver's lane change. Karatas *et al.* [25] used head-mounted inertial sensors for head tracking in vehicles. Yang *et al.* [20] presented an acoustic ranging system to locate the smartphone on vehicle using its audio infrastructure. Wang *et al.* [19] captured vehicle dynamics with OBD devices, and compared with smartphone sensing to determine the position of the smartphone in vehicle. However, these works rely on extra devices, which may not be widely available and thus reduce the practical usefulness of the approaches for being quickly adopted among a large number of users. Instead, our GazMon is the first eye gazing based active driving behavior monitoring and prediction framework that can be implemented with the COTS mobile hardware, making GazMon a promising real-life deployment that can benefit many applications to actively improve driving safety.

In the driving mobile applications, the driving behavior recognition has attracted a lot of research efforts, such as Augmented Driving[12] and CarSafe [26]. Augmented Driving provides lane changing assistance and safe following distance based on the vision of driver and captured by smartphone camera. CarSafe [26] detects and alerts drivers to dangerous driving conditions and behavior based on the smartphone camera. Chen *et al.* [27] employed inertial sensors to detect various driving behavior, including lane changes, turns, and driving on curvy roads. Karatas *et al.* [28] tracked steering wheel usage and angle with a wearable watch and smartphone. Recently, eye-gazing has been studied in the context of driving, aiming at predicting the drivers' next moves by relying on their eye fixations [29]. When driver's gaze cannot be directly acquired through eye tracking systems, Vicente *et al.* [30] inspected drivers' faces using landmarks and the head orientation. Takatsugu *et al.* [31] proposed a classifier to determine the cognitive distraction and neutral states with the driver's gaze transition. In industry, the safety systems work based on near-IR methods. Lexus has equipped their high-end LS models with their Driver Attention Monitor,[13] which is designed to detect whether a driver is not looking forward and will signal an alert if it detects an object ahead. The system permanently monitors the movement of the driver's head when looking from side to side using a near-IR camera installed on the top of the steering wheel column, which is also integrated into the pre-crash system, so as to warn the driver when a collision is

11. https://tobiigaming.com/eye-tracker-4c/

12. http://imaginyze.com/Site/Welcome.html
13. http://www.lexus.com/models/LS/safety

probable. Different from these approaches, our work strives to construct a deep learning architecture to predict the driving behavior that is going to happen based on various preprocessed eye gaze information, which, can allow more time gap beyond the two-second rule and bring great potentials to significantly improve the driving safety.

8 Conclusion

In this paper, we presented GazMon that can predict driving behaviors based on the drivers' gaze patterns. GazMon employs an image preprocessing scheme to extract eye gazing features, which potentially offers the rich information for driving behavior prediction. We then construct a deep learning architecture by utilizing the Convolutional Neural Network and Long Short Term Memory network to effectively solve the driving behavior prediction problem. A prototype has been implemented using Android smart phones, and our extensive experimental results have demonstrated that GazMon achieves the driving behavior prediction accuracy of 94 percent on average in daily driving environments, which is better than the state-of-art machine learning approaches and provides a viable framework to allow many applications to benefit from predicting the driving behaviors.

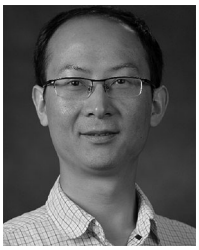
Acknowledgments

This research was supported by NSERC Postdoctoral Fellowship and an NSERC Discovery Grant. Part of Xiaoyi's work was also supported by the National Natural Science Foundation of China 61602214.

References

[1] S. Karush, "Crashes avoided: Front crash prevention slashes police-reported rear-end crashes," *Insurance Inst. Highway Saf. Status Rep.*, vol. 51, no. 1, Jan. 28, 2016.
[2] T. A. Dingus *et al.*, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proc. Nat. Academy Sci. United States America*, vol. 113, no. 10, pp. 2636–2641, 2016.
[3] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1867–1874.
[4] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.*, vol. 15, no. 1, pp. 123–141, 1995.
[5] Y.-M. Cheung and Q. Peng, "Eye gaze tracking with a web camera in a desktop environment," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 4, pp. 419–430, Aug. 2015.
[6] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L 0 gradient minimization," *ACM Trans. Graphics*, vol. 30, 2011, Art. no. 174.
[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[9] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.
[10] *Learn to Drive Smart*. Insurance Corporation of British Columbia, 2015. [Online]. Available: https://www.icbc.com/driver-licensing/Documents/driver-full.pdf
[11] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Berlin, Germany: Springer, 2008.
[12] K. Krafka *et al.*, "Eye tracking for everyone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2176–2184.
[13] S. Hu *et al.*, "Experiences with eNav: A low-power vehicular navigation system," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 433–444.
[14] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.
[15] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 656–667.
[16] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2033–2046, Oct. 2014.
[17] J.-G. Wang, E. Sung, and R. Venkateswarlu, "Estimating the eye gaze from one eye," *Comput. Vis. Image Understanding*, vol. 98, no. 1, pp. 83–103, 2005.
[18] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 802–815, Feb. 2012.
[19] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proc. ACM 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 41–54.
[20] J. Yang *et al.*, "Detecting driver phone use leveraging car speakers," in *Proc. ACM 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 97–108.
[21] C. Bo, X. Jian, X.-Y. Li, X. Mao, Y. Wang, and F. Li, "You're driving and texting: Detecting drivers using personal smart phones by leveraging inertial sensors," in *Proc. ACM 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 199–202.
[22] M. Sodhi, B. Reimer, J. Cohen, E. Vastenburg, R. Kaars, and S. Kirschenbaum, "On-road driver eye movement tracking using head-mounted devices," in *Proc. Symp. Eye Tracking Res. Appl.*, 2002, pp. 61–68.
[23] B. Qi, P. Liu, T. Ji, W. Zhao, and S. Banerjee, "DrivAid: Augmenting driving analytics with multi-modal information," in *Proc. IEEE Veh. Netw. Conf.*, 2018, pp. 1–8.
[24] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 453–462, Sep. 2009.
[25] C. Karatas, H. Li, and L. Liu, "Toward detection of unsafe driving with inertial head-mounted sensors," in *Proc. 8th Wireless Students, by Students Students Workshop*, 2016, pp. 45–47.
[26] C.-W. You *et al.*, "CarSafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones," in *Proc. ACM 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 13–26.
[27] D. Chen, K.-T. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible sensing of vehicle steering with smartphones," in *Proc. ACM 13th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2015, pp. 1–13.
[28] C. Karatas *et al.*, "Leveraging wearables for steering and driver tracking," in *Proc. IEEE INFOCOM*, 2016, pp. 1–9.
[29] N. Pugeault and R. Bowden, "How much of driving is preattentive?," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5424–5438, Dec. 2015.
[30] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2014–2027, Aug. 2015.
[31] T. Hirayama, K. Mase, C. Miyajima, and K. Takeda, "Classification of driver's neutral and cognitive distraction states based on peripheral vehicle behavior in driver's gaze transition," *IEEE Trans. Intell. Veh.*, vol. 1, no. 2, pp. 148–157, Jun. 2016.

Xiaoyi Fan received the BE degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and the MSc and PhD degrees from Simon Fraser University, Burnaby, Canada, in 2015 and 2018, respectively. He is currently an honorary postdoctoral research fellow with the Department of Electrical and Computer Engineering, University of British Columbia. He is a recipient of the Chinese Government Award for Outstanding Self-finance Students Abroad (2017) and the NSERC Postdoctoral Fellowship (2019). His research interests include the Internet of Things, edge computing, and deep learning. He is a member of the IEEE.



Feng Wang (S'07–M'13–SM'18) received the bachelor's and master's degrees in computer science and technology from Tsinghua University, Beijing, China, in 2002 and 2005, respectively, and the PhD degree in computing science from Simon Fraser University, Burnaby, British Columbia, Canada, in 2012. He is currently an associate professor with the Department of Computer and Information Science, University of Mississippi, University, Mississippi. He is a recipient of IEEE ICME Quality Reviewer Award (2011). He is a Technical Committee member of the *Elsevier Computer Communications*. He served as program vice chair in International Conference on Internet of Vehicles (IOV) 2014, and as TPC co-chair in IEEE CloudCom 2017 for Internet of Things, and Mobile on Cloud track. He also serves as TPC member in various international conferences such as IEEE INFOCOM, ICPP, IEEE/ACM IWQoS, ACM Multimedia, IEEE ICC, IEEE GLOBECOM, and IEEE ICME. He is a senior member of the IEEE.

Danyang Song received the BSc and BEng degrees from the dual degree program of Zhejiang University, Hangzhou, Zhejiang, China, and Simon Fraser University, Burnaby, Canada, in 2018. He is currently working toward the master's degree at Simon Fraser University, Burnaby, Canada. His research interests include edge computing, cloud computing, system, and networking. He is a student member of the IEEE.

Yuhe Lu received the bachelor's degree in computing science from Simon Fraser University, Burnaby, BC, Canada and Zhejiang University, Hangzhou, Zhejiang, China, in 2017. He is currently working toward the master's degree in applied science from the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. His research interests include edge computing and software engineering. He is a student member of the IEEE.

Jiangchuan Liu (S'01–M'03–SM'08–F'17) received the BEng (Cum Laude) degree from Tsinghua University, Beijing, China, in 1999, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, in 2003, both in computer science. He is currently a full professor (with University professorship) with the School of Computing Science, Simon Fraser University, British Columbia, Canada. He is a steering committee member of the *IEEE Transactions on Mobile Computing*, and associate editor of *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Big Data*, and *IEEE Transactions on Multimedia*. He is a co-recipient of the Test of Time Paper Award of IEEE INFOCOM (2015), ACM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and ACM Multimedia Best Paper Award (2012). He is a fellow of the IEEE, a fellow of Canadian Academy of Engineering and an NSERC E.W.R. Steacie Memorial fellow.