Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach

Yang Xing, Chen Lv, *Member, IEEE*, Huaji Wang, Dongpu Cao, *Member, IEEE*, Efstathios Velenis, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—Driver decisions and behaviors are essential factors that can affect the driving safety. To understand the driver behaviors, a driver activities recognition system is designed based on the deep convolutional neural networks (CNN) in this paper. Specifically, seven common driving activities are identified, which are the normal driving, right mirror checking, rear mirror checking, left mirror checking, using in-vehicle radio device, texting, and answering the mobile phone, respectively. Among these activities, the first four are regarded as normal driving tasks, while the rest three are classified into the distraction group. The experimental images are collected using a low-cost camera, and ten drivers are involved in the naturalistic data collection. The raw images are segmented using the Gaussian mixture model to extract the driver body from the background before training the behavior recognition CNN model. To reduce the training cost, transfer learning method is applied to fine tune the pre-trained CNN models. Three different pre-trained CNN models, namely, AlexNet, GoogLeNet, and ResNet50 are adopted and evaluated. The detection results for the seven tasks achieved an average of 81.6% accuracy using the AlexNet, 78.6% and 74.9% accuracy using the GoogLeNet and ResNet50, respectively. Then, the CNN models are trained for the binary classification task and identify whether the driver is being distracted or not. The binary detection rate achieved 91.4% accuracy, which shows the advantages of using the proposed deep learning approach. Finally, the real-world application are analyzed and discussed.**

***Index Terms*—Driver behavior, driver distraction, convolutional neural network, transfer learning.**

Manuscript received February 20, 2018; revised June 24, 2018, November 1, 2018, January 31, 2019, and March 20, 2019; accepted March 25, 2019. Date of publication April 1, 2019; date of current version June 18, 2019. This work was supported in part by the Young Elite Scientist Sponsorship Program by CAST under Grant 2017QNRC001 and in part by the SUG-NAP of Nanyang Technological University, Singapore, under Grant M4082268.050. The review of this paper was coordinated by Dr. A. Chatterjee. *(Corresponding authors: Chen Lv and Dongpu Cao.)*

Y. Xing and C. Lv are with the School of Mechanical and Aerospace Engineering, Nanyang Technology University, Singapore 639798 (e-mail: yxing_edu@163.com; lyuchen@ntu.edu.sg).

H. Wang is with AVL Powertrain UK Ltd, CV4 7EZ Coventry, U.K. (e-mail: Huaji.wang@avl.com).

D. Cao is with the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: dongpu.cao@uwaterloo.ca).

E. Velenis is with the Advanced Vehicle Engineering Centre, Cranfield University, MK43 0AL Bedford, U.K. (e-mail: e.velenis@cranfield.ac.uk).

F.-Y. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: feiyue@ieee.org).

Digital Object Identifier 10.1109/TVT.2019.2908425

I. Introduction

A. Motivations

DRIVER is in the center of the Road-Vehicle-Driver (RVD) loop. Driver decision and behaviors are the major aspects that can affect driving safety. It is reported that more than 90% light vehicle accidents are caused by human driver misbehavior in the United States, and the accident rate can be reduced by 10% to 20% with a precise driver behavior monitoring system [1]–[5]. Therefore, the recognition of driver behaviors is becoming one of the most important tasks for intelligent vehicles. For the conventional advanced driver assistance systems (ADAS), the driver is in the center of the RVD loop. The understanding of driver behavior enables the ADAS to generate the optimal vehicle control strategies which is suitable for the current driver states [6]–[8]. Regarding the intelligent and highly automated vehicles, such as the Level-3 automated vehicles (according to the definition in Society of Automotive Engineers standard J3016), the driver is responsible for taking over the vehicle control under emergencies. At this moment, the real-time driver behavior and activity monitoring system has to decide whether the driver can take over or not.

Therefore, in this study, a deep learning-based driver activities recognition system is proposed to monitor and understand the driver behaviors continuously. The recognition models are trained to identify seven common driving-related tasks and also to determine whether the driver is being distracted or not. With this end-to-end approach, intelligent vehicles can better interact with human drivers and properly making decisions and generating human-like driving strategies.

B. Related Works

Driver behaviors have been widely studied over the past two decades. Previous studies mainly focus on the driver attention [41] and distraction (either physical distraction or cognitive distraction) [36], driver intention [7], [9], driver styles [42], driver drowsiness and fatigue detection [10]–[12], etc. The National Highway Traffic Safety Administration (NHTSA) defined driver distraction as a process that the driver shifts their attention away from the driving tasks. Four types of distraction are clarified by the NHTSA, which are the visual distraction, auditory distraction, biomechanical distraction, and cognitive distraction [37]. To understand the driver behaviors, most of the studies require capturing the driver status information, such as the head pose

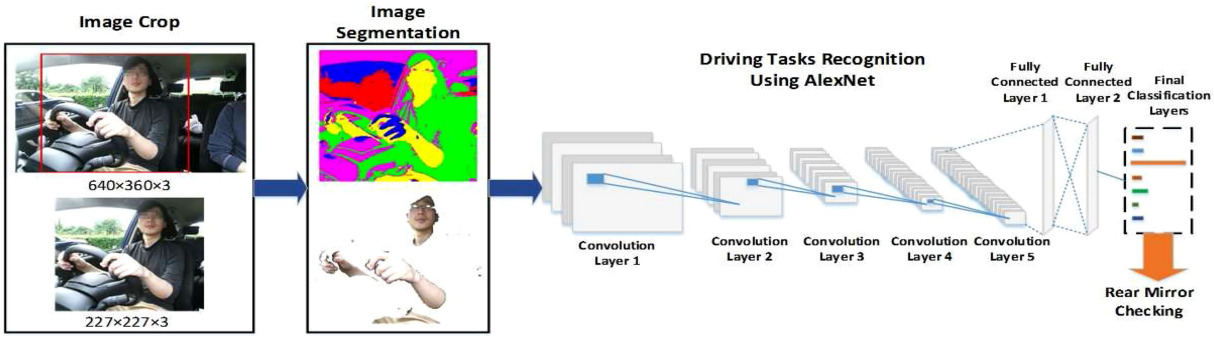


Fig. 1. Overall system architecture.

[13], eye gaze [8], hand motion [14], foot dynamics [15], and even the physiological signals [16], [17].

Specifically, in [18], the video information for the driver head movement along with the audio signals was collected to identify the secondary driving tasks. In [19], the drivers head pose, eye gaze direction, and hand movement were combined to identify driver activities. In [20], the driver's head poses estimation was proposed and applied to the rear-end crash avoidance system. Despite the vision-based feature extraction methods, the physiological signals, such as the electroencephalogram (EEG) and electrooculography (EOG) are also widely used for real-time driver status monitoring. In [21], EEG signals were collected to predict the driver braking intention. In [22], the EEG and EOG signals were used to estimate the driver drowsiness and fatigue status. The EEG signals are proved to be closely related to the driver behaviors and can illustrate an earlier response to the human mental states compared with the outer physical behaviors.

However, as aforementioned, most of the existing driver behavior studies require extracting specific features in advance, such as the head pose angle, gaze direction, EEG, and the position of hand and body joints [23]. These features are not always easy to be obtained, and some even require specific hardware devices, which will increase either the temporal or the financial cost. Therefore, in this work, an end-to-end driver activity recognition system is proposed based on the deep CNN models, which is accurate and easy to be implemented. To study the driver distraction behaviors, visual distraction, auditory distraction, and biomechanical distraction are involved. While the cognitive distraction is not considered since it has been well studied in [38], [39], which can be effectively detected with a non-vision-based approach.

Regarding the current development of deep learning techniques, significant progress has been made in the computer vision area dues to the development of deeper CNN models, parallel computing hardware, and the large-scale annotated dataset. Deep CNN models have achieved the state-of-art results in object detection, classification, generation, and segmentation tasks. Meanwhile, it has been successfully applied to some driver monitoring tasks [24], [25]. In this work, three different CNN models will be evaluated for driver activities recognition and distraction detection tasks. The only sensor required in this study is a low-cost RGB camera. Based on the report in [26], seven most common in-vehicle activities for both manual driving and automated driving vehicles are selected, which contains normal driving activities as well as secondary tasks. The CNN models take the processed images directly without any manual feature extraction procedure. By applying the transfer learning scheme, the pre-trained CNN models can be efficiently fine-tuned to satisfy the behaviors detection task.

C. Contribution

The contribution of this study can be summarized as follows. First, a novel deep learning-based approach is applied to identify driver behaviors. Unlike existing studies that require complex algorithms to estimate the driver status information, the proposed algorithm takes merely the color images as the input and directly outputs the driver behavior information. With the deep CNN models, the manually feature extraction process can be replaced by an automatic feature learning process.

Second, transfer learning is applied to fine-tune the pre-trained deep CNN models. The models are trained to deal with both the multiple classification tasks and the binary classification task. The algorithm is proved as a practical solution for non-intrusive driver behavior detection. Besides, this study also shows that transfer learning can successfully transfer the domain knowledge that learned from the large-scale dataset to the small-scale driver behavior recognition task.

Finally, an unsupervised GMM-based segmentation method is applied to process the raw images and extract the driver body region from the background. It is found that by applying a segmentation model prior to the behavior detection network, the detection accuracy on the driving activities recognition can increase significantly.

D. Paper Organization

The remainder of this paper is organized as follows. Section II introduces the experiment setup and data collection. Section III proposes the deep convolutional neural network models and transfer learning schemes for driving tasks recognition. Then, the tasks recognition results and model evaluation are performed in Section IV. Section V presents the discussions and future works. Finally, this paper is concluded in Section VI.

II. Experiment and Data Collection

This section describes the experimental design and data processing for driver behavior recognition. Fig. 1. illustrates the

Fig. 2. Experiment setup. The Kinect is mounted on the middle of the front window and data are collected using a laptop.

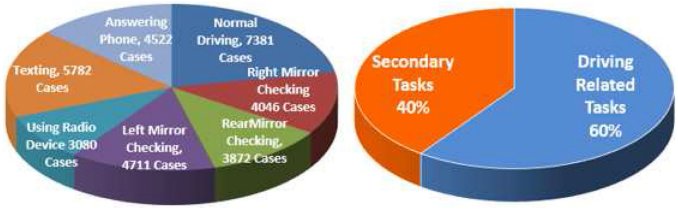


Fig. 3. Illustration of the collected dataset. The left part shows the number of images for each tasks. The right part shows the general data distribution between the normal driving group and the secondary tasks.

general system architecture. First, raw RGB images are collected using the Kinect camera. Then, the cropped images are segmented using the GMM algorithm. Finally, the CNN model is adopted for the activities recognition task. Specifically, driver behavior images are collected with a Kinect camera. The Kinect enables the collection of multi-modal signals, such as the color image, depth image, and audio signals. It was initially designed for indoor human-computer-interaction and has been successfully used for driver monitoring systems [27], [28]. As mentioned in [23], the drivers' head poses, and upper body joints also can be detected using the Kinect. While in this study, only the RGB images are used.

According to the Kinect application requirements [30], it was mounted in the middle of the front window, facing the upper body of the driver so that not to interfere the drivers field of view while driving. The device setup is shown in Fig. 2. The sampling rate for the image collection is 25 frames per seconds. According to the study in [18], short-term driver behaviors like mirror checking can last from 0.5 to 1 second. Therefore, the sampling rate is fast enough to capture these behaviors. The data are recorded with an Intel Core i7 2.5 GHz CPU, and the codes are written in C++ based on the Windows Kinect SDK and OpenCV. To store the images, the raw images are compressed to $640 \times 360 \times 3$ format to increase the computation efficiency.

Ten drivers are involved in the experiment. They were asked to perform seven activities, which consist of four normal driving tasks (normal driving, left mirror checking, right mirror checking, and rear mirror checking) and three secondary tasks (using in-vehicle radio/video device, answering mobile phone, and texting). It took about 20 to 30 minutes for each driver to finish all these tasks, and about 34 thousands images were captured in total. In this experiment, five drivers were asked to perform these tasks during driving in a testing field, while the rest five drivers were asked to mimic the driving tasks and not drive the vehicle. This is because, during normal driving, it is dangerous to perform the secondary tasks so that secondary data are limited. However, the steady scenario can be used to collect enough secondary behavior data safely. The number of images for each task and the quantitative comparison between normal driving and secondary tasks is shown in Fig. 3. Unlike some human activity recognition studies that require temporal information, in this study, no temporal information is considered, and each image is processed individually. The reason is that during driving the driver outer behaviors are always explicit, such as mirrors checking and performing secondary tasks. Therefore, no temporal information is required for inferring driver behaviors since most of the images carry enough information for activity recognition.

III. Methodologies

This section describes the algorithm framework that is used in this study. Specifically, Section III-A introduces the image pre-processing and segmentation based on the GMM algorithm. Section III.B describes the three deep CNN frameworks as well as the transfer learning scheme.

A. Image Pre-Processing and Segmentation

The original images are stored in the format of $640 \times 360 \times 3$. The raw images are cropped to speed up the CNN training process and increase the classification accuracy. An interest of region (ROI) which mainly contains the driver body region is selected. The left part of Fig. 1. indicates the raw image and the selected ROI. After the raw images are cropped, these images are transformed into the size of $227 \times 227 \times 3$ to satisfy the input requirement of the AlexNet and $224 \times 224 \times 3$ for the GoogLeNet and ResNet, respectively.

Then, the GMM algorithm is applied to segment the images and extract the driver body region from the background. GMM is an unsupervised machine learning method, which can be used for data clustering and data mining. It is a probability density function that is represented by a weighted sum of sub-Gaussian components [30]. One of the advantages of using GMM to unsupervised segment the images is it requires no manual labeling and can be flexible to modify the model by adjusting the cluster centers [31]. To train a GMM-based segmentation model, each image is represented by a feature vector according to the pixel intensity. The feature vector for the GMM is a three-dimensional vector that contains the RGB intensity of each pixel.

Fig. 4. illustrates the segmented images of the ten drivers for model training and testing. Driver head and body region can be identified with the GMM segmentation method. Since the camera is fixed inside the vehicle cabin, the drivers seat position and the corresponding head position will be fixed within a certain area. The driver body region can be determined based on a set of pre-defined points which are located around the drivers head

Fig. 4. Illustration of the raw images and segmented images.

position. The points around the head position and the corresponding label will be used to indicate the driver regions. In the future, the manual selection method can be replaced by using an automatic detection method. For example, a precise driver head position can be first detected using the head detection algorithms and then the driver body regions can be determined directly or using a simple semantic segmentation network. As shown in the next section, the segmentation-based method can dramatically increase the model recognition accuracy.

B. Model Preparation and Transfer Learning

1) AlexNet Model: Currently, deep convolutional neural networks have gained a tremendous improvement in the domain of computer vision. One of the key reasons is the distribution of ImageNet dataset [32]. ImageNet is a large-scale dataset, which contains more than 15 million high-resolution annotated natural images of over 22,000 categories. A large number of annotated images benefit the training of deeper and more accurate CNN models. In this work, three deep convolutional neural network models, namely, AlexNet, GoogLeNet, and ResNet50 are chosen as the basic model structures for the recognition of driver behavior. The AlexNet was first proposed by Alex Krizhevsky in 2012 [33], who won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC12). The model was trained for the classification of 1000 categories in the ImageNet dataset. There are five convolutional layers and three fully connected neural network layers with non-linearity and pooling layers between the convolutional layers. In total, AlexNet contains 60 million parameters and 650,000 neurons. An simplified model structure for AlexNet is shown in Fig. 1.

2) GoogLeNet Model: GoogLeNet is another deep CNN model, which won the ILSVRC14 [40]. GoogleNet is significantly deeper than the AlexNet, and it achieved more accurate classification results on the ImageNet dataset. Despite the model depth, the main contribution of GoogLeNet is the utilization of Inception architecture. As shown in [40], the most common ways

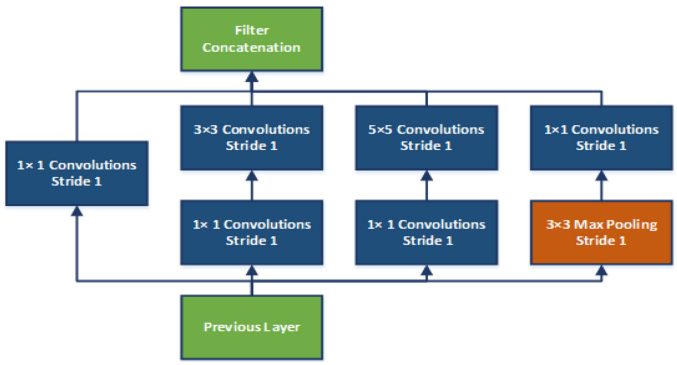


Fig. 5. Inception layer of GoogLeNet. There are six convolution filers and one max-pooling filter within each Inception layer [40].

of increasing CNN model performance are to improve the network size (either the depth or the width of the model). However, it gives rise to the requirement for larger scale dataset and more computational burden. Based on this, the Inception layers was introduced into the CNN model to increase the sparsity among the layers, and reduce the number of parameters. Each Inception layer consists of six basic convolution filters and one max pooling filter. With different scales, the parallel-arranged convolutional filters will have more accurate detailing and a broader representation for the information from previous layers. A typical dimension reduction Inception layer is shown in Fig. 5. In total, there are two traditional convolutional layers at the lower level of the GoogLeNet and nine Inception layers are concatenated at higher levels. With the application of Inception layers, the general quantity of the parameters in GoogLeNet is 12 times less than that in the AlexNet.

3) ResNet Model: Recent evidence has shown that the network depth is of importance to the feature representation and generalization [44]. It is common to see that simply stack the convolutional layers to increase the depth of the model cannot give a better training and generalization performance [45].

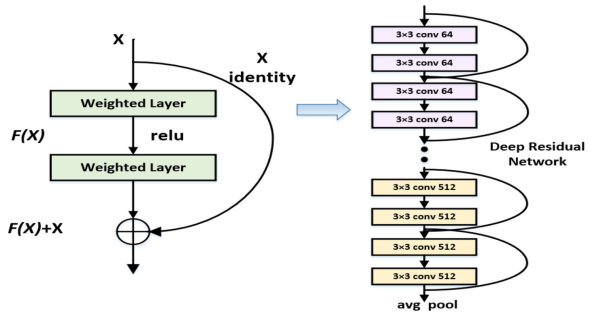


Fig. 6. Residual learning block and deep residual network [43].

Accordingly, in [43], Kaming *et al.* introduced a novel deep CNN model, namely, Residual Networks (ResNet) to enable the construction of deeper convolutional neural networks. By introducing the residual learning scheme, the ResNet achieved the first place on the ILSVRC 2015 classification competition and won the ImageNet detection, ImageNet localization, COCO detection 2015, and COCO segmentation.

As shown in the left part of Fig. 6, the underlying mapping function for the basic residual block can be assumed as H(x). The x represents the inputs to the first layer. The residual network supposes an explicit residual mapping function F(x) exists such that F(x)=H(x)-x, and the original mapping can be represented as F(x)+x. The core idea behind the residual network block is that although both H(x) and the F(x)+x mapping is able to approximate the desired functions asymptotically, it is much easier to learn the mapping of F(x)+x. The added layers through the shortcut connection are the identity mapping. The right graph in Fig. 6 represents the full structure of a deep residual network, where residual learning is performed for every few stacked layers. By introducing the identity mapping and copying the other layers from the shallower model, the deep residual network can efficiently solve the model degradation problem when the models getting deeper [43].

4) Transfer Learning: A large-scale annotated dataset like ImageNet is needed to train the deep convolutional neural networks like AlexNet, GoogLeNet, and ResNet from scratch. However, in general, large-scale annotated datasets are not always available for specific tasks. Therefore, the common ways to use the pre-trained deep CNN model are either treating the model as a fixed feature extractor without tuning the model parameters or fine-tune the pre-trained model parameters with a small-scale dataset. In this study, the CNN models will be used in the second manner, which is to fine tune the last few layers of the models with the driver behavior dataset. Since the original models are trained to classify the 1000 categories, the last few layers have to be modified so that the models can satisfy the seven objects or the binary classification task. Specifically, the original last fully connected layer and the output layer, which generate the probabilities for the 1000 categories, are replaced by a new fully-connected layer and softmax layer that output the probabilities for the seven categories.

The basic structure and properties of the convolutional layers is remained so that these layers can keep their advantages in the feature extraction and representation. Meanwhile, the knowledge that learned from the large-scale ImageNet dataset can be transferred to the driver behavior domain. A small initial learning rate is selected to slow down the updating rate of the convolutional layers. On the contrary, a much larger learning rate for the last fully connected (FC) layer is chosen to speed up the learning rate in the final layers. In this study, the convolutional layers are not frozen as we found that the performance will decrease when totally freezing the convolutional layers. Therefore, a small updating rate was chosen so that the convolutional layer will try to adapt to the new classification task. With this kind of combination, the new models can be trained to solve the new classification tasks.

TABLE I
CLASSIFICATION RESULTS FOR DRIVING TASKS RECOGNITION USING ALEXNET

No.	GMM Based AlexNet							
	T1	T2	T3	T4	T5	T6	T7	Ave
D1	0.825	0.929	0.011	0.225	0.840	1.0	0.972	0.771
D2	0.875	0.234	0.571	0.229	0.516	0.928	0.836	0.813
D3	0.564	0.684	0.0	0.711	0.747	0.983	0.983	0.908
D4	0.825	0.469	0.927	0.399	0.0	0.958	0.994	0.786
D5	0.797	0.20	0.10	0.843	0.60	0.959	0.996	0.843
D6	0.957	0.928	0.852	0.977	0.783	0.926	0.999	0.928
D7	0.993	0.921	0.915	0.951	0.913	0.290	0.981	0.878
D8	0.990	0.989	0.417	1.0	0.991	0.996	0.736	0.880
D9	0.353	0.994	0.229	0.813	1.0	0.982	0.979	0.752
D10	0.528	0.724	0.447	0.798	0.274	1.0	0.995	0.684
Mean	0.786	0.869	0.545	0.802	0.771	0.932	0.945	**0.816**

IV. EXPERIMENT RESULTS AND ANALYSIS

In this section, the analysis for the driving activities classification are proposed. The system performances are evaluated from four major aspects: the impact image segmentation on multi-behaviors recognition, deep CNN model visualisation, the binary classification results on the distracted behavior detection, and the performance comparison with other methods.

A. Evaluation of CNN Models on the Multiple Driving Behaviors Recognition

In this section, the activities recognition results for the ten participants are evaluated. The seven driving-related tasks are ordered as normal driving, right mirror checking, rear mirror checking, left mirror checking, using radio/video device, texting, and answering mobile phone. Table I, Table II, and Table III illustrate the classification results of the seven tasks based on AlexNet, GoogLeNet, and ResNet, respectively. T1 to T7 represents the seven tasks and D1 to D10 indicates the ten different drivers. The models are trained with MATLAB Deep Learning toolbox and evaluated using the *leave-one-out (LOO)* cross-validation method. To get the activity identification results for each driver, the images from one driver are used as testing images, whereas the rest images of the nine drivers are used for training. Therefore, for each driver, the data are completely new

TABLE II
CLASSIFICATION RESULTS FOR DRIVING TASKS RECOGNITION USING GOOGLENET

No.	GMM Based GoogLeNet							
	T1	T2	T3	T4	T5	T6	T7	Ave
D1	0.917	0.619	0.0	0.325	0.433	1.0	0.968	0.768
D2	0.892	0.362	0.0	0.042	0.230	0.784	0.815	0.767
D3	0.883	0.563	0.0	0.073	0.840	1.0	0.994	0.739
D4	0.740	0.453	0.848	0.986	0.758	0.663	1.0	0.755
D5	0.970	0.20	0.233	0.325	0.078	0.959	0.988	0.799
D6	0.951	0.966	0.807	0.936	0.967	0.075	1.0	0.829
D7	1.0	0.886	0.436	0.990	0.890	0.248	0.963	0.737
D8	0.301	0.995	0.178	1.0	1.0	0.990	0.998	0.789
D9	0.562	0.245	0.949	0.997	1.0	0.990	0.843	0.792
D10	0.990	1.0	1.0	0.685	0.882	0.012	1.0	0.810
Mean	0.835	0.766	0.648	0.796	0.819	0.678	0.948	**0.786**

TABLE III
CLASSIFICATION RESULTS FOR DRIVING TASKS RECOGNITION USING RESNET50

No.	GMM Based ResNet50							
	T1	T2	T3	T4	T5	T6	T7	Ave
D1	0.944	0.389	0.120	0.125	0.219	1.0	0.963	0.746
D2	0.872	0.284	0.0	0.729	0.066	0.918	0.926	0.921
D3	0.919	0.938	0.195	0.040	0.814	0.998	0.993	0.753
D4	0.975	1.0	0.924	0.514	1.0	0.639	0.882	0.801
D5	0.907	0.255	0.133	0.874	0.473	0.930	0.996	0.856
D6	0.790	0.992	0.941	0.791	0.504	0.509	0.985	0.750
D7	0.996	0.857	0.629	0.922	0.950	0.301	0.973	0.786
D8	0.528	0.567	0.192	0.641	0.988	0.944	0.715	0.638
D9	0.346	0.245	0.713	0.997	0.735	0.693	0.829	0.655
D10	0.002	0.999	0.058	0.991	0.782	0.219	1.0	0.589
Mean	0.728	0.652	0.391	0.662	0.653	0.715	0.926	**0.749**

to the CNN models and the identification performances equal the model generalization on this new dataset.

As shown in Table I, the general identification accuracy for the segmentation-based AlexNet achieved an average of 81.4% accuracy. The raw-image based AlexNet was also tested, which achieved only 69.2% recognition accuracy. In Table I, the average performance in the rightmost column is defined as the average detection results for each driver, while the mean accuracy in the bottom row represents the average detection rate for each task. Regarding the detection accuracy for each task, the answering mobile phone activity gets the most accurate detection results among the ten drivers for all three models. The worst result happens in the rear mirror checking (T3) case for the three models. One explanation is that the rear mirror checking behavior require few body and head movement, which can be easily misclassified into the normal driving task. Another evidence that can be drawn from Table I, Table II, and Table III are the CNN model achieved better detection results on the secondary tasks in general. This is mainly because when performing the secondary tasks, the driver has to move his/her body and hands instead of

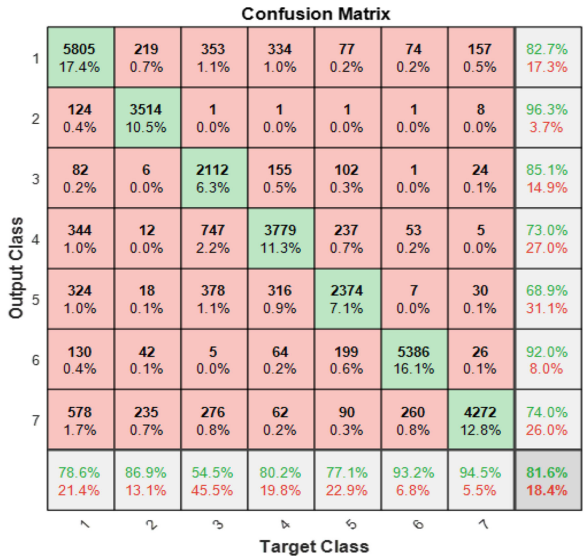


Fig. 7. Confusion matrix of secondary tasks detection with AlexNet.

Fig. 8. Illustration of texting behavior of driver 3. The left image is the raw image, and the right image is the segmented image.

only rotating her/his head, which is more distinct and easier to be detected.

Table II indicates the activity classification results given by the GoogLeNet. The general detection results is similar to the results in Table I except that the overall detection accuracy for the ten drivers are slightly lower. The GoogLeNet does not achieve better classification results than the AlexNet as it does on the ImageNet dataset. However, the classification results for the GoogLeNet trained with raw images are better than that in the AlexNet case. The general classification results for the GoogLeNet with the raw image is 74.7% accuracy, which is 5% higher than that for the AlexNet. Table III illustrates the activity classification results given by the ResNet. Same to the GoogLeNet, the ResNet does not show its advantage on the activity classification task. Instead, the precision is the lowest among these three models. The general classification accuracy is 74.9% for the GMM-ResNet and 61.4% for the Raw image-based ResNet. Discussions on the results will be proposed in the next section.

Fig. 7. illustrates the confusion matrix for the ten drivers using the AlexNet model with GMM segmentation. The green diagonal shows the correct detection cases for the class. The bottom row shows the classification accuracy with respect to the target class, while the rightmost column shows the classification accuracy with respect to the predicted labels. As shown in Fig. 7,

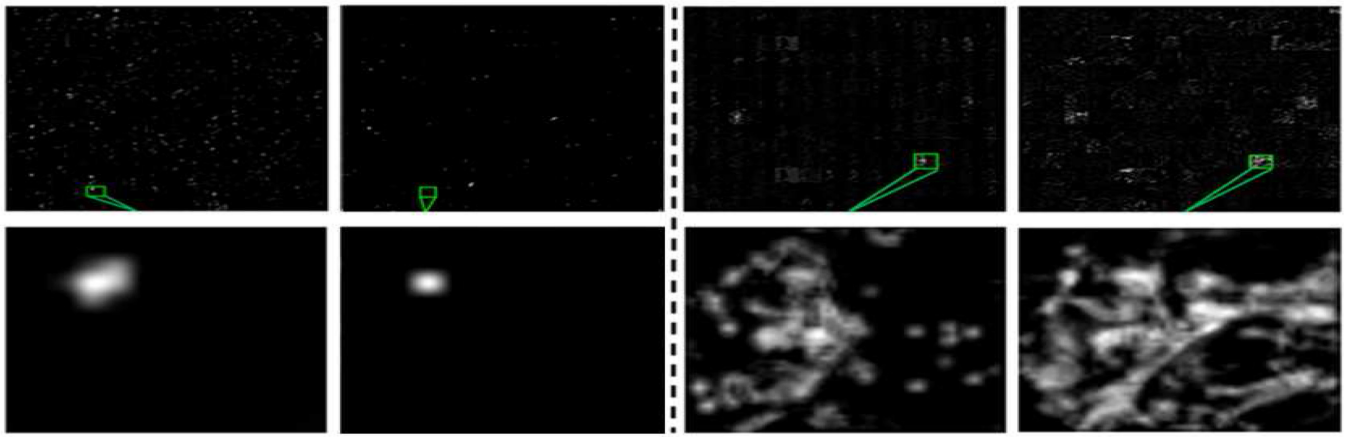


Fig. 9. Activation map of the two CNN models. The left part is given by the AlexNet model and the right part is given by the GoogLeNet. From left to right, the Relu5 layer activation of the GMM-AlexNet model, the Relu5 layer of the Raw-AlexNet model, the conv2-Relu layer of the GMM-GoogLeNet, and the same layer activation map for the Raw-GoogLeNet model. The lower part images are the corresponding strongest activation channels in the activation maps for the GMM-AlexNet and GMM-GoogLeNet models.

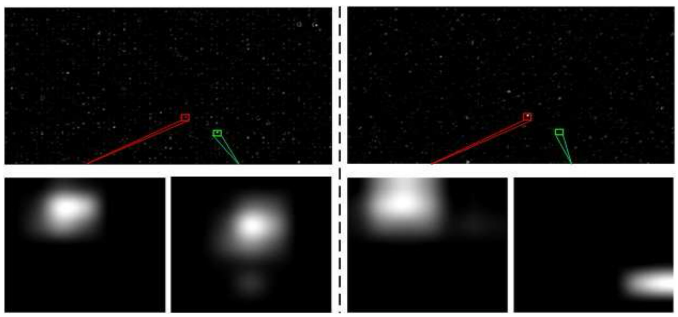


Fig. 10. Activation map of the ResNet. The left part indicates the activation of the final Relu layer for the G-ResNet, while the right part is for the R-ResNet. The red boxes in the upper images are the strongest activation for the Raw-ResNet, while the green boxes represent the strongest activation for the G-ResNet. The lower four images are the corresponding activation channels.

all of the driving tasks except the third task (rear mirror checking) achieved reasonable detection rates. There are 353 cases of rear-mirror checking are misclassified into normal driving and 747 cases are misclassified into the left mirror checking.

B. Visualisation of Deep CNN Models

To further understanding of how the model responses to the segmented images, the activation map and feature visualization for the CNN models are analyzed. Fig. 8. shows the raw image and the segmented image of in-vehicle texting behavior for one participants. The image pair will be used to generate the model activation map.

Fig. 10. shows the activation maps of AlexNet and GoogLeNet with respect to the images shown in Fig. 8. As shown in the left part of the dashed line in Fig. 10, the top activation maps are given by the Relu5 layer of the AlexNet models. Specifically, the top left one is based on the AlexNet with GMM segmentation (G-AlexNet model), whereas the second one is based on the AlexNet with raw images (R-AlexNet model). The bottom images are the corresponding strongest activation channel for the segmentation-based model. As shown in the left part of Fig. 10, the Relu5 layer for the G-AlexNet model remains much more features than that in the R-AlexNet. The segmentation-based method can extract the driver from the background more precisely so that the CNN model can maintain more relevant features of the driver. The strongest activation of the G-AlexNet model keeps the driver head rotation and other channels can store the arm position information as well.

Since the GoogLeNet is much deeper than the AlexNet, only the activation map of the second convolutional layer is analyzed. The right part of Fig. 10. indicates the activation of the conv2-ReLu layer in the GoogLeNet. As shown in the figure, the G-GoogLeNet trained with segmented images carries more driver related features instead of background features. As the driver-related features are not well maintained in the beginning layers, the deeper Inception layers of the R-GoogeLeNet also cannot learn very representative features for the driver. Based on this, the activation maps explain why the segmentation-based CNN models lead to a better classification result than the raw image-based models.

Similar results can be found in the ResNet case which is shown in Fig. 10. The upper images show the activation maps for the G-ResNet and R-ResNet, respectively. The lower part indicates the corresponding strongest activation map. Specifically, the green box in the top row images is the strongest activation channel of the final ReLu layer of the G-ResNet, and its corresponding channel in the R-ResNet map. The red boxes in the top images are the strongest activation channel of the R-ResNet, and its corresponding channel in the G-ResNet. As can be seen in the lower part of Fig. 10, the strongest activation channel of the G-ResNet is able to capture the head rotation and position features, while the R-ResNet fails to learn a precise representation for this behavior. Based on the model visualization results, it can be seen that with a prior image segmentation, the CNN model can learn more representative driver status features.

Finally, the time cost for model training and testing are compared in Table IV. As shown in Table IV, the general training

TABLE IV
TRAINING AND TESTING TIME COST FOR EACH MODEL

	G-ANet	R-ANet	G-GNet	R-GNet	G-RNet	R-RNet
Training Time (s)	∼1100	∼1200	∼2400	∼2400	∼3600	∼3600
Testing per Image (ms)	∼13	∼12.5	∼45	∼45	∼140	∼140

for the GoogLeNet is two times longer than the AlexNet, and it takes about one hour to train the ResNet on the local computational device. It takes about 12 ms to process one image for the AlexNet while the testing time for the GoogLeNet and ResNet are 45 ms and 140 ms, respectively. The model training and testing are implemented on an Intel Core i7 2.5 GHz CPU and NVIDIA MX150 2 GB GPU.

C. Driver Distraction Detection Using Binary Classifier

In this section, the three CNN models are modified and trained to detect whether the driver is distracted or not. In this case, the first four tasks are grouped together, while the last three tasks constitute another group. The CNN models are fine-tuned to solve the binary classification problem. The distraction detection results for the AlexNet, GoogLeNet, and ResNet are shown in Table V. As shown in Table V, the segmentation image based AlexNet leads to the most accurate results. The general classification accuracy for the G-AlexNet based model is 91.4%. The general classification accuracy for the GoogLeNet and ResNet methods are 87.5% and 83.0%, which are slightly lower than the results given by the AlexNet. It should be noticed that there are no smoothing algorithms applied to the distraction warning module. In real-world situations, the driver assistance system will only warn the driver if the distraction happens continuously in a short period. Therefore, if applying a short period smoothing or voting techniques, the distraction detection system can be more suitable for the real-world application.

D. Comparison With Other Methods

To further evaluate our method, additional experiments are made to compare the proposed method with conventional handcraft feature extraction and shallow CNN methods. Specifically, the approaches used for comparison include:

FC7+ANN: The method proposed in [34], which extracts the posture features with a pre-trained AlexNet CNN model. In this part, the activation of 'fc7' layer of AlexNet is extracted, and an FFNN ANN model with 300 neurons is constructed based on the feature set. The dimension for each of the 'fc7' feature vector is 4096.

PHOG+SVM: The pyramid histogram of oriented gradients (PHOG) followed by support vector machine (SVM) method. A pyramid HOG feature extractor, which concatenates two different scale HOG extractors is used. The block size for the HOG feature extractors is 22, and the cell sizes are 88 and 1616, respectively. The dimension for each of the PHOG feature vector is 32328.

OP+ANN: The method proposed in [51], [52], which recognize the motion with optical flow. Specifically, the optical flow of the video sequence is extracted with the Lucas-Kanade method, and a 51529-dimensional feature vector is concatenated for each image. Then, another FFNN model with the same structure of the one that proposed in [23] is used.

OPsCNN: Based on the magnitude of the optical flow, a shallow multi-class CNN is proposed. Three convolution layers are used following with three fully connected layers. The input images are rescaled into the size of 120×120. The filter size is selected as 5×5 for the first two convolutional layers, and 3×3 for the third convolutional layer. Batch normalization, non-overlap pooling, and ReLu non-linearity layers are applied between the convolutional layers. The number of neurons for the three FC layers are 512, 128, and 7, which is similar to the architecture used in [50].

GMMsCNN: The shallow CNN with the GMM segmented color images are also tested. Finally, as the dimensions of the feature vectors given by different algorithms are too high, a principal component analysis (PCA) algorithm is used to reduce the feature dimension and reduce the training cost for the first three models. The dimension for each feature vector is reduced to 500. The model comparisons are illustrated in Table VI. As shown in Table VI, the PHOV and optical flow features are unable to accurately represent the driving tasks and far less precise than the transfer learning method. The recognition results of the optical flow-based and shallow CNN-based methods are slightly better than the feature extraction methods. The high-level features from the FC7 layer of AlexNet with FFNN gives better results than the rest four methods. However, the average results for the ten drivers are still significantly lower compared with the proposed method.

In Table VII, the proposed method is also compared with relevant studies in the literature. It should be noticed that difficulties exist in making a precise cross-platform comparison between the existing studies since different algorithms, platforms, and experimental methods were used. Based on Table VII, some researchers have tried to analysis the driver distracted behaviors with either real vehicle and simulated data. For example, in [18], Li *et al.* proposed a machine learning framework for the detection of driver mirror checking behaviors and secondary tasks. The general framework follows a standard machine learning application procedure, which consists of feature extraction, model training, and testing. The detection rates for secondary tasks like radio operating and phone-talking are around 75% to 80%. We believe this should attribute to the absent of driver body features. In [23], the driver's behaviors are detected with a Kinect device, which enables the analysis of both driver's head and upper body features. However, that work also heavily rely on the complex feature extraction and analysis, which is time-consuming and requires extra hardware for calibration. Similar work can be found in [50], where the authors evaluated the performance of different types of CNN models on ten different driving activities. Although high detection accuracies are achieved in the study, the data are collected on the driving simulator and did not stand for the real-world in-vehicle performance. Another reason that can significantly influence the model accuracy is the evaluation

TABLE V
BINARY CLASSIFICATION RESULTS USING ALEXNET

		D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Mean
Binary AlexNet	Normal	0.970	0.994	0.849	0.873	0.991	0.897	0.897	0.997	0.911	1.00	0.936
	Distract	0.910	0.673	0.858	0.917	0.763	0.856	0.856	0.941	0.948	0.988	0.881
	Ave	0.948	0.867	0.852	0.903	0.856	0.882	0.882	0.976	0.927	0.996	**0.914**
Binary GoogLeNet	Normal	0.993	0.940	0.141	0.850	0.988	0.992	0.992	0.833	0.908	0.999	0.897
	Distract	0.784	0.577	1.00	0.870	0.857	0.692	0.847	0.884	0.898	0.934	0.841
	Ave	0.916	0.796	0.426	0.863	0.911	0.883	0.946	0.853	0.904	0.978	**0.875**
Binary ResNet50	Normal	0.908	0.950	0.656	0.609	0.989	0.820	0.982	0.871	0.06	0.985	0.798
	Distract	0.905	0.768	0.524	0.975	0.794	0.941	0.946	0.809	1.00	0.994	0.891
	Ave	0.907	0.878	0.612	0.852	0.874	0.864	0.971	0.847	0.955	0.988	**0.830**

TABLE VI
ACTIVITY RECOGNITION COMPARED WITH OTHER APPROACHES

	T1	T2	T3	T4	T5	T6	T7	Mean
FC7+ANN	0.478	0.343	0.113	0.249	0.311	0.803	0.631	0.497
PHOG+SVM	0.573	0.059	0.024	0.394	0.108	0.437	0.473	0.354
OP+ANN	0.404	0.033	0.093	0.121	0.209	0.561	0.506	0.347
OPsCNN	0.537	0.369	0.085	0.273	0.205	0.572	0.669	0.443
GMMsCNN	0.423	0.242	0.096	0.109	0.212	0.598	0.531	0.400
Proposed	0.786	0.869	0.545	0.802	0.771	0.932	0.945	**0.816**

method. In Table VII, the *Loo* method is more strict than the *cross validation* method as it indicates the model generalization capability on the unseen dataset. If we use the cross-validation method and simply separate the data into training and testing group, the GMM-AlexNet in this study can achieve 98.9% accuracy for multi-tasks detection. However, we still suggest using the *Loo* method as it can reflect the performance variance on different subjects.

Based on the comparison with existing studies, the proposed method in this study show three advantages. First, a naturalistic in-vehicle dataset is collected for the fine-tuning and validation of the deep CNN models. The fine-tune method is very efficient in real-world application as it is hard to collect large scale annotated driver distraction data. Although some studies use side view images as the images show clear driver body features [47]–[50], the side view method is less efficient and robust compared with the front view method in the real vehicle as the side view can be occluded by the passengers. Second, the leave-one-out model evaluation is used so that the results illustrate an independent performance on the different drivers. Third, the segmentation-based CNN models achieved state-of-art detection accuracy on distracted behaviors such as the phone answering (93.2%) and texting (94.5%) with naturalistic data.

V. DISCUSSION

A. Transfer Learning Performance

With the analysis of different deep CNN models, it can be found that deeper CNN model like ResNet50 does not contribute to a higher detection accuracy as it did in the ILSVRC competition. The reasons can be multifold, and we try to explain this phenomenon merely based on the evidence in this study. First, as the GoogLeNet and ResNet are deeper than the AlexNet, the model may need more data to be optimized. Second, the transfer learning approach is different from training from scratch such as using the ImageNet dataset. The fine-tune transfer learning method mainly focus on the tuning of a few layers while keeping the main characters of the convolutional layers. However, as the model getting deeper, the domain knowledge learned from the much larger dataset may not very suitable for the smaller dataset. This conclusion is only made according of the results in this study and further evaluation is expected. The primary object in this study is not to evaluate the classification performance of different CNN models. However, this study aims to provide an efficient end-to-end approach to understand driver behaviors. Therefore, more experiment and analysis are expected in the future to obtain a more precise explanation.

Although it is essential to understand which features are more critical to the driver behavior recognition, the traditional machine learning framework is less efficient and usually has a specific requirement on the system hardware such as head pose measurement and skeleton tracking. Therefore, in this work, an end-to-end deep learning approach is designed to solve the driver behavior recognition task. The only hardware required for the proposed system is an RGB camera. Meanwhile, as shown in this study, the CNN models can automatically capture the head and body features. The deep learning method achieved competitive detection results compared with the methods that rely on the head and body detection [23], [46]. However, the end-to-end process shows its advantages in real-world detection since no complex head pose and body joint estimation algorithms are needed. Besides, this study also evaluated the binary classification results and found that the deep learning approach can provide an accurate estimation of the distraction status. This approach can be easily integrated into most of the current ADAS products dues to its efficient and low-cost properties.

B. Real-Time Application

In this experiment, the system is implemented to a Windows operating system using MATLAB platform and a single low-cost GPU device. The testing cost of the AlexNet for each image is

TABLE VII
CLASSIFICATION RESULTS USING FEATURE EXTRACTION

ID	Inputs	Tasks	Model	Validation	Platform	Subjects	Computation Cost
[24]	Kinect RGB and depth, body, head, eye	7 tasks: 4 mirror checking, radio, phone call, texting	Random Forest and FFNN	*Loo*, Recognition: 82.4%	Real Vehicle	5 drivers	8 fps data collection
[47]	Kinect RGB and depth image (Eye,arm,head,and facial features)	5 tasks:phone call,drinking,message,looking object,normal driving	Sequential model with AdaBoost and HMM	*Loo*, Recognition: 85.0%, Detection: 89.8%	Simulator	8 drivers	1 fps screenshot of the monitors
[48]	Triple-view fusion	7 tasks:gear, driving, phone call, phone pick, control, looking left/right	CNN+RNN sequential feature extractor with SVM classifier	*Cross Validation*, 90% in average	Simulator	3 drivers	15 fps data collection
[19]	CAN and cameras with 268D features	4 mirror tasks and radio,GPS operating and following,phone operating and call, picture,coversation	SVM, KNN, RUSBoost	*Loo*, recognition rates for different tasks are among 65%-85%	Real Vehicle	20 drivers	5s window size with 10 samples per window
[49]	Side view images with face and hand detection	10 tasks: dringking, radio, normal driving, makeup, reach behind, conversation, phone call, texting	Transfer learning with AlexNet and Inception V3	*CV*, 75%training data and 25%testing, 95.98% in average	Real Vehicle	31 drivers	AlexNet 182 fps and Inception V3 72 fps with GTX TITAN
[50]	Side view images	10 tasks: dringking, radio, normal driving, makeup, reach behind, conversation, phone call, texting	Transfer learning with VGG16, AlexNet, GoogLeNet, and ResNet	*Loo*, recognition accuracy in the range of 86% and 92%	Simulator	10 drivers	Frequency in the range of 8 and 14Hz with Jetson TX1
[51]	Side view images	4 tasks: normal driving, Operating shift gear, phone call, eating/smoking	Sparse filter and CNN model	*CV*, 80%training and 20%testing, 99.47% in average	Real Vehicle	20 drivers	-
Ours	Front view images	7 tasks: 4 mirror checking, radio, phone call, texting	GMM segmentation and transfer learning	*Loo*, Recognition: 81.6%, Detection: 91.4%	Real Vehicle	10 drivers	14 fps with Nvidia MX150 GPU

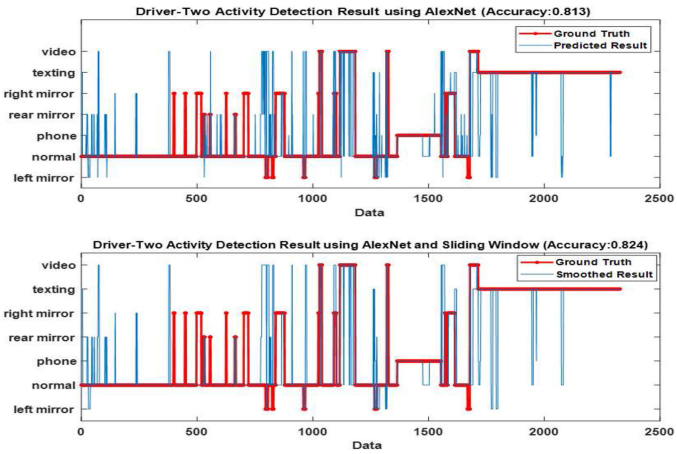


Fig. 11. Driver 2 real-time activity detection using AlexNet and sliding-window.

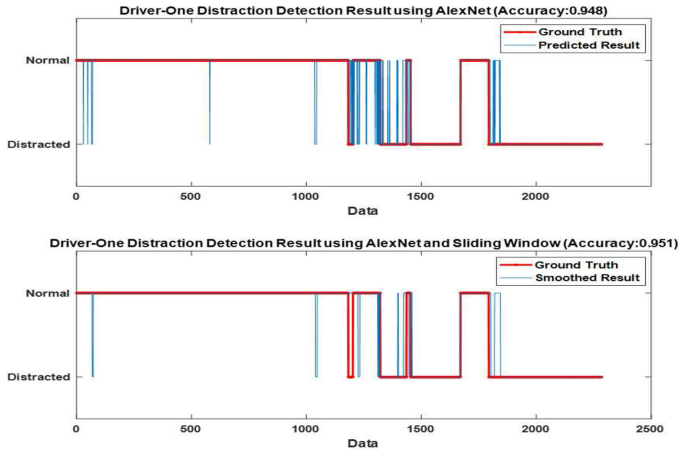


Fig. 12. Driver 1 real-time activity detection using AlexNet and sliding-window.

about 13 ms, also, it cost 50 ms for the GMM to segment each image. The total computational cost for each image is around 60–70 ms, and the general processing ability of the system is about 14fps. Therefore, the proposed system can satisfy the real-world computational requirement. Meanwhile, regarding the in-vehicle embedded systems in the real-world, the Linux platform along with C++/Python programming usually can be more efficient than the MATLAB environment. Also, since more powerful embedded GPU devices has been published, the in-vehicle graphics processor can provide more powerful parallel computation than the current platform. Hence, the algorithm has no significant limitation in the real-world application.

Next, a sliding-window is applied to the detection to smooth the result. The current driver state is selected according to the majority state within the sliding window. The smoothing results of the secondary tasks detection and distraction detection are shown in Fig. 12 and Fig. 13. The upper images of Fig. 12 and Fig. 13 indicate the comparison between the ground truth label and the predicted values concerning the seven driving activities detection and distraction detection, while the bottom images represent the comparison between the ground truth label and the smoothed version of the predicted values. The sliding window can be used to smooth the result, and eliminate some false detection cases. However, it should be noticed that as the sliding-window uses the voting scheme, detection delay can happen

for the secondary tasks and the horizon of the sliding-window will control how much delay the detection system has. A larger horizon of the sliding-window will lead to a smoother result; however, it will also cause a larger detection delay. In Fig. 12 and Fig. 13, the window is selected as seven samples, which can cause a 500 ms delay. Considering each task, especially the secondary tasks can last several seconds, this 0.5 s late detection is normally acceptable.

VI. Conclusion

In this work, a driving-related activity recognition system based on the deep CNN model and transfer learning method is proposed. To increase the identification accuracy, the raw RGB images are first processed with a GMM-based segmentation algorithm, which can efficiently remove the irrelevant objects and identify the driver position from the background context. The classification results indicate that the segmentation contributes to a much more precise detection result than the model trained with the raw images. Another comparison is made between the transfer learning and other feature extraction methods. Finally, if using the CNN models as a binary classifier, the driver distraction detection rate can achieve 91% accuracy. In the future, the data will be further analyzed, and the model will be updated to increase the system robustness and detection accuracy. Meanwhile, the system will be tested and used for driver/passenger behavior analysis on the partially automated vehicles in the real world.

References

[1] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in CPSS: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 577–587, Oct. 2017.

[2] J. Wang, J. Wang, R. Wang, and C. Hu, "A framework of vehicle trajectory replanning in lane exchanging with considerations of driver characteristics," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3583–3596, May 2017.

[3] A. Koesdwiady, R. Soua, F. Karray, and M. S. Kamel, "Recent trends in driver safety monitoring systems: State-of-the-art and challenges," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4550–4563, Jun. 2017.

[4] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, "Capturing car-following behaviors by deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 910–920, Mar. 2018.

[5] X. Zeng and J. Wang, "A stochastic driver pedal behavior model incorporating road information," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 5, pp. 614–624, Oct. 2017.

[6] C. Lv *et al.*, "Analysis of autopilot disengagements occurring during autonomous vehicle testing," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 58–68, Jan. 2018.

[7] V. A. Butakov and P. Ioannou, "Personalized driver/vehicle lane change models for ADAS,"*IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4422–4431, Oct. 2015.

[8] C. Gou *et al.*, "A joint cascaded framework for simultaneous eye detection and eye state estimation," *Pattern Recognit.*, vol. 67, pp. 23–31, 2017.

[9] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse Bayesian learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 431–440, Sep. 2007.

[10] J. Hu, L. Xu, X. He, and W. Meng, "Abnormal driving detection based on normalized driving behavior," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6645–6652, Aug. 2017.

[11] R. Chai *et al.*, "Driver fatigue classification with independent component by entropy rate bound minimization analysis in an EEG-based system," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 715–724, May 2017.

[12] B. Mandal *et al.*, "Towards detection of bus driver fatigue based on robust visual analysis of eye state," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 545–557, Mar. 2017.

[13] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, Jun. 2010.

[14] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 2953–2958.

[15] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Comput. Vis. Image Understanding*, vol. 116, no. 3, pp. 435–445, 2012.

[16] L. Bi, Y. Lu, X. Fan, J. Lian, and Y. Liu, "Queuing network modeling of driver EEG signals-based steering control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 8, pp. 1117–1124, Aug. 2017.

[17] T. Teng, L. Bi, and Y. Liu, "EEG-Based detection of driver emergency braking intention for brain-controlled vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1766–1773, Jun. 2018.

[18] N. Li and C. Busso, "Detecting drivers' mirror-checking actions and its application to maneuver and secondary task recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 980–992, Apr. 2016.

[19] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, "Head, eye, and hand patterns for driver activity recognition," in *Proc. 22nd Int. Conf. IEEE Pattern Recognit.*, 2014, pp. 660–665.

[20] M. Rezaei and R. Klette, "Look at the driver, look at the road: No distraction! No accident!," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2104, pp. 129–163.

[21] I. H. Kim, J.-W. Kim, S. Haufe, and S.-W. Lee, "Detection of braking intention in diverse situations during simulated driving based on EEG feature combination," *J. Neural Eng.*, vol. 12, no. 1, 2014, Art. no. 016001.

[22] C. Zhang, W. Hong, and R. Fu, "Automated detection of driver fatigue based on entropy and complexity measures," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 168–177, Feb. 2014.

[23] Y. Xing *et al.*, "Identification and analysis of driver postures for in-vehicle driving activities and secondary tasks recognition," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 1, pp. 95–108, Mar. 2018.

[24] T. H. N. Le, C. C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "DeepSafeDrive: A grammar-aware driver parsing approach to Driver Behavioral Situational Awareness (DB-SAW)," *Pattern Recognit.*, vol. 66, pp. 229–238, 2017.

[25] L. Li, W.-L. Huang, Y. Liu, N.-N. Zheng, and F.-Y. Wang, "Intelligence testing for autonomous vehicles: A new approach,"*IEEE Trans. Intell. Vehicles*, vol. 1, no. 2, pp. 158–166, Jun. 2016.

[26] M. Sivak and B. Schoettle, "Motion sickness in self-driving vehicles," Univ. Michigan Transp. Res. Inst., Ann Arbor, MI, USA, Rep. UMTRI-2015-12, 2015.

[27] S. Gaglio, G. Lo Re, and M. Morana, "Human activity recognition process using 3-D posture data," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 5, pp. 586–597, Oct. 2015.

[28] L. B. Neto *et al.*, "A Kinect-based wearable face recognition system to aid visually impaired users," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 1, pp. 52–64, Feb. 2017.

[29] M. Azimi, "Skelton joint smoothing white paper," *Tech. Report*, Microsoft Inc, Redmond, WA, USA, Tech. Rep. 2012.

[30] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc.Adv. Neural Inf. Process. Syst*., 2000, pp. 554–560.

[31] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Comput.*, vol. 8, no. 1, pp. 129–151, 1996.

[32] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

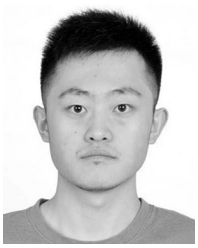
[34] N. Deo, A. Rangesh, and M. Trivedi, "In-vehicle hand gesture recognition using hidden Markov models," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 2179–2184.

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 886–893.

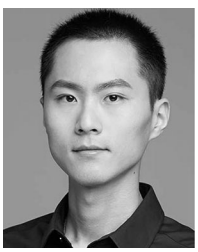
[36] A. Tawari, S. Sivaraman, M. M. Trivedi, T. Shannon, and M. Tippelhofer, "Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking," in *Proc. IEEE Intell. Vehicles Symp.*, 2014, pp. 115–120.

[37] T. A. Ranney, W. R. Garrott, and M. J. Goodman, "NHTSA driver distraction research: Past, present, and future," SAE Int., Warrendale, PA, USA, Tech. Rep. 2001-06-0177, 2001.

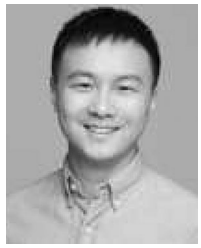
[38] Y. Liao, S. E. Li, W. Wang, Y. Wang, G. Li, and Bo Cheng, "Detection of driver cognitive distraction: A comparison study of stop-controlled intersection and speed-limited highway,"*IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1628–1637, Jun. 2016.
[39] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1108–1120, Apr. 2016.
[40] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[41] N. Pugeault and R. Bowden, "How much of driving is preattentive?," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5424–5438, Dec. 2015.
[42] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, Mar. 2018.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556.
[45] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, arXiv:1505.00387.
[46] C. Craye and F. Karray, "Driver distraction detection and recognition using RGB-D sensor," 2015, arXiv:1502.00250.
[47] R. Kavi, V. Kulathumani, F. Rohit, and V. Kecojevic, "Multiview fusion for activity recognition using deep neural networks," *J. Electron. Imag.*, vol. 25, no. 4, 2016, Art. no. 043010.
[48] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," 2017, arXiv:1706.09498.
[49] D. Tran, H. M. Do, W. Sheng, H. Bai, and G. Chowdhary, "Real-time detection of distracted driving based on deep learning," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1210–1219, Dec. 2018.
[50] C. Yan, B. Zhang, and F. Coenen., "Driving posture recognition by convolutional neural networks," *IET Comput. Vis.*, vol. 10, no. 2, pp. 103–114, Mar. 2016.
[51] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3192–3199.
[52] L. Sevilla-Lara, Y. Liao, F. Guney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," in *Proc. German Conf. Pattern Recognition (GCPR 2018)*, pp. 281–297, 2018.



Yang Xing received the M.Sc. with distinction in control systems from the University of Sheffield, Sheffield, U.K., in 2014 and the Ph.D. degree in transport systems from Cranfiled University, Bedford, U.K., in 2018. He is currently a Research Fellow with the Nanyang Technological University, Singapore. His research interests include driver behavior modeling, driver-vehicle interaction, and advanced driver assistance systems.



Chen Lv received the Ph.D. degree from Tsinghua University, Beijing, China in 2016. He is currently an Assistant professor at Nanyang Technological Univerity, Singapore. From 2014 to 2015, he was a Joint Ph.D. Researcher with EECS Deptartment, University of California, Berkeley, CA, USA. His research interests include cyber-physical system, hybrid system, advanced vehicle control, and intelligence, where he has contributed over 40 papers and obtained 11 patents in China. Dr. Lv serves as a an Associate Editor for *International Journal of Electric and Hybrid Vehicles*, *International Journal of Vehicle Systems Modelling and Testing*, *International Journal of Science and Engineering for Smart Vehicles*, and *Journal of Advances in Vehicle Engineering*.



Huaji Wang received the B.S. degree in automotive engineering from Jilin University, Changchun, China, in 2005, and the Ph.D. degree in engineering from the University of Cambridge, Cambridge, U.K., in 2016, concentrating on the study of driver/vehicle systems, and driver-automation collaboration. He is currently working as a System Engineer in automated driving with the AVL Powertrain, Coventry, U.K.

Dongpu Cao received the Ph.D. degree from Concordia University, Canada, in 2008. He is currently an Associate Professor and Director of Waterloo Cognitive Autonomous Driving (CogDrive) Lab at the University of Waterloo, ON, Canada. He has contributed more than 180 publications, two books, and one patent. His current interests include driver cognition, automated driving, and cognitive autonomous driving. Dr. Cao was a recipient of the SAE Arch T. Colwell Merit Award in 2012, and three best paper awards from the ASME and IEEE conferences. He is a Canada Research Chair in Driver Cognition and Automated Driving.He serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and *ASME Journal of Dynamic Systems, Measurement and Control*. He was a Guest Editor for *Vehicle System Dynamics* and IEEE TRANSACTIONS ON SMC: SYSTEMS.

Efstathios Velenis received the Ph.D. degree from the School of Aerospace Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, GA, USA, in 2006. He is currently a Senior Lecturer with the Advanced Vehicle Engineering Centre, Cranfield University, Cranfield, U.K. He obtained research funding from EPSRC, Innovate U.K., and the European Commission. His current research interests include vehicle dynamics and control, optimal control for active chassis systems, traction, braking, and handling control for electric/hybrid vehicles, lap-time optimization, and tire modeling. Dr. Velenis was a recipient of the Luther Long Award for best Ph.D. dissertation in the area of engineering mechanics from Georgia Tech.

Fei-Yue Wang (S'87–M'89–SM'94–F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined the University of Arizona, Tucson, AZ, USA in 1990 and became a Professor and Director of the Robotics and Automation Laboratory (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program from the State Planning Council and "100Talent Program" from CAS, and in 2002, he was appointed as the Director of the Key Lab of Complex Systems and Intelligence Science, CAS. From 2006 to 2010, he was the Vice President of Research, Education, and Academic Exchanges at the Institute of Automation, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems. His current research interests include methods and applications for parallel systems, social computing, and knowledge automation.