Accident Analysis and Prevention 137 (2020) 105432

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Detection of driver manual distraction via image-based hand and ear recognition

Li Li[a], Boxuan Zhong[b], Clayton Hutmacher Jr[a], Yulan Liang[c], William J. Horrey[d], Xu Xu[a,*]

[a] Edward P. Fitts Department of Industrial & Systems Engineering, North Carolina State University, Raleigh, NC 27695, United States
[b] Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606, United States
[c] TrueMotion, Boston, MA 02210, United States
[d] AAA Foundation for Traffic Safety, Washington, DC 20005, United States

ARTICLE INFO

Keywords:
Driving distraction
Upper extremity kinematics
Deep learning
Computer vision
Multi-class classification

ABSTRACT

Driving distraction is a leading cause of fatal car accidents, and almost nine people are killed in the US each day because of distracting activities. Therefore, reducing the number of distraction-affected traffic accidents remains an imperative issue. A novel algorithm for detection of drivers' manual distraction was proposed in this manuscript. The detection algorithm consists of two modules. The first module predicts the bounding boxes of the driver's right hand and right ear from RGB images. The second module takes the bounding boxes as input and predicts the type of distraction. 106,677 frames extracted from videos, which were collected from twenty participants in a driving simulator, were used for training (50%) and testing (50%). For distraction classification, the results indicated that the proposed framework could detect normal driving, using the touchscreen, and talking with a phone with F1-score 0.84, 0.69, 0.82, respectively. For overall distraction detection, it achieved F1-score of 0.74. The whole framework ran at 28 frames per second. The algorithm achieved comparable overall accuracy with similar research, and was more efficient than other methods. A demo video for the algorithm can be found at https://youtu.be/NKclK1bHRd4.

1. Introduction

1.1. Motivation

Driving distraction is defined as "the delay in the recognition of information needed to safely accomplish the driving task because some event, activity, object, or person within or outside the vehicle compels or induces the driver's shifting attention away from the driving task" (Ranney, 2008). It is the leading cause of fatal car crashes. According to the National Highway Traffic Safety Administration (NHTSA), over 8% percent of fatal crashes are distraction-affected. The number of fatalities in distraction-affected crashes in 2017 was 3166. In other words, almost nine people are killed each day because of distracting activities. Therefore, reducing the number of distraction-affected traffic accidents remains an imperative issue. For that reason, establishing a methodology for monitoring the driver's status is critical for improving driving safety. Furthermore, acquiring the inattention status of the driver is also an essential part of the co-pilot system to determine the driving mode (Kim et al., 2016).

In a naturalistic driving study of adolescent drivers, a range of distracting behaviors were summarized. These behaviors include the driver: holding a cell phone to his or her ear, taking on a hands-free phone, operating on electronic devices or suspected of operating electronic devices, adjusting the controls of the vehicle, grooming, eating or drinking, reaching for an object in-vehicle, communicating with someone outside vehicle, turning around, reading, etc. (Foss and Goodwin, 2014). According to a study of the driver's engagement of distracting behaviors (Stutts et al., 2005), manipulating music/audio controls is the most common in-vehicle distracting activity among drivers (91.4%); smoking (includes lightning and extinguishing) takes up the most of time in potentially distracting activities during vehicle moving (21.1%). Other top-ranked activities among drivers include drinking/eating/spilling (71.4%), grooming (45.7%), reading or writing (40%) and using cell phone/pager (34.3%). Therefore, detecting the most frequent non-driving activities is crucial to the establishment of the driver's monitoring system.

1.2. Detecting distraction through image tracking

Distracting activities cause excess manual, visual, and cognitive

* Corresponding author.
E-mail address: xxu@ncsu.edu (X. Xu).

https://doi.org/10.1016/j.aap.2020.105432
Received 14 June 2019; Received in revised form 2 January 2020; Accepted 3 January 2020
Available online 28 January 2020

demands (Fernández et al., 2016). In most cases, a single non-driving task will involve at least two or three kinds of increased demands, e.g. using the touchscreen during driving will lead to both extra visual and manual demands. While cognitive demands associated with these distractions can be difficult to observe, most non-driving tasks can be detected through exploiting information from the driver's response to the increased manual or visual demands, because many common distractions involve different kinds of overt features like the driver's hand movement, or gaze direction, etc. (Stutts et al., 2005). For example, the event “hands off the wheel” can be a universal indicator for the start of distracting activities; the event “right hand and right ear get very close” can be used to infer the cell phone use. Thus, through applying vision-based algorithms, the driver's status can be inferred from those visually observable features.

1.3. Related work

Previous research has used combined signals of human responses to the manual, visual, and cognitive demands induced by distractions. In general, they fall into two primary categories.

The first one mainly exploits facial features combined with other physiology signals, which typically include eye closure, eyelids, blink rate, gaze direction, eye saccadic movement, mouth opening size, and head movements (Sigari et al., 2013, 2014; Sigari, 2009). A previous study (Liang et al., 2007) used eye movement and measures of driving data (steering wheel angle, lane position, and steering error transformed from steering wheel angle) to train a Support Vector Machine (SVM) for detecting distraction. Another study (Miyaji et al., 2009) adopted eyes, head movement and electrocardiogram (ECG) data as input, and used adaboost as the classifier. Two types of distraction, having conversation and doing mental arithmetic, were detected at an accuracy of 89.8% and 90.3% respectively. Eye-tracking is also one of the most frequently used techniques Hurtado and Chiasson (2016). A very recent study Schwarz et al. (2019) used a driver monitoring system (DMS) placed on the steering column for classifying observational ratings of drowsiness (ORDs), and achieved 82% accuracy in classifying three levels of drowsiness based on ORDs ratings (from being awake to severely drowsy).

The major limitation for these methods is that the predictive outputs are often binary. It is hard to distinguish what kind of distraction occurs simply from the driver's eye gaze or facial expressions. Furthermore, while facial expressions encode rich emotional and intentional information Majumder et al. (2016), one or more cameras need to be placed in front of the face (e.g. 40–50 cm if placed on the dashboard) for satisfactory resolution (Gokturk et al., 2002; Fridman, 2018; Vicente et al., 2015). The corresponding algorithms are also subject to more environmental challenges, such as a constantly changing illumination level (Zhang et al., 2017).

The second category extracts contextual information primarily from in-vehicle image input combined with other secondary signals like lane keeping data for detection. Image data can be acquired simply through installing an in-vehicle video camera, which is non-intrusive to drivers and low-cost. Moreover, the contextual information is more generalizable compared to personalized facial features.

A previous study (Wollmer et al., 2011) trained a Long Short-Term Memory (LSTM) deep neural network on head movement and lane tracking data, and achieved 91.6% in accuracy for two-class classification, and 43.3% for six-class classification. Another research (Ngan Le et al., 2016) proposed a multi-scale faster regional convolutional neural network (faster-RCNN) to detect the driver's hands, wheel and cell phone simultaneously. The event “cell phone intersects with hand” indicated cell phone use. The event “hand off the wheel” became a universal indicator for non-driving tasks. While they achieved good accuracy in detecting both events, a potential limitation was the overall efficiency. The reported frame per second (FPS) was 0.09, which is not likely to be sufficient for a real-time application.

Manual distractions are generally associated with one or more body segments and joints, such as upper arms, lower arms and hands. The hand usually serves as the end effector during distracting tasks. Previous research successfully used its location relative to other body segments for recoginizing manual distraction (Gallahan et al., 2013). In our previous study (Li et al., 2019), spatial data of driver's right hand collected from inertial measurement units (IMUs) proved to be effective for inferring different types of distraction, including drinking, texting, talking on a cell phone, using touchscreen and placing a marker in a cup holder. However, the results indicated that talking on a cell phone is the least discernible event. Because of similar spatial patterns of the right hand among different events, the algorithm was less sensitive in distinguishing making a phone call from using the touchscreen and texting. Therefore, more spatial features are needed for a more robust detection algorithm.

In this paper, we propose a novel image-based driver distraction recognition method by detecting and situating the driver's right hand and right ear. The distractions include drinking, texting, talking on a cell phone, using touchscreen and placing a marker in a cup holder. The detection of the right ear will further assist in classifying different distracting events, especially for making a phone call where a driver needs to hold the phone by the ear. The method consists of two major modules. The first module incorporates You Only Look Once (YOLO), a fast and robust deep neural network for object detection. It addresses the issue of low FPS. The second module takes the coordinates of regions of interest (ROIs) of the ear and hand as input, and a multi-layer perceptron is designed to infer the driver's status from the ROIs. This framework provides a prototype for detecting more types of distraction. To evaluate the performance of this framework, video data of five different types of distracting activities was collected in a driving simulator.

2. Method

2.1. Overview

The proposed method consists of two modules (Fig. 1). In the first module, YOLO, an object detection deep neural network (DNN), was adopted. It takes images as input and detects the driver's right ear and right hand simultaneously. The second module is a multi-layer perceptron, which takes the ROIs as input and outputs the predicted type of distraction.

In a basic object detection task, the input is an image with or without target objects, and the output is the bounding box of each target object, which is formulated as:

$$(x, y, w, h) \tag{1}$$

The notation is illustrated in Fig. 2.

Ideally, the output bounding boxes are supposed to cover the regions of interest (ROIs). For a common object detection task, an object can be categorized into a certain class. However, the categorization of an object is dependent on the training set rather than the actual division of natural objects. In this paper, the neural network was trained only to detect the driver's right hand and right ear, so other similar objects like left hand or left ear were regarded as negative examples. We also included prior knowledge during detection, which assumes that only the right hand and the right ear of a driver were visible in the camera view, so for each class we took the bounding box with the highest confidence score as the final classification.

The first module detects the driver's right hand and right ear and outputs their bounding boxes. As mentioned before, each bounding box can be formulated as a 4-D vector, and so two 4-D vectors were concatenated into a single 8-D vector. Each convolutional neural network can be regarded as a regression function that takes an image as input and projects it to a fixed-dimensional vector defined previously. It follows that module one can be formulated as:

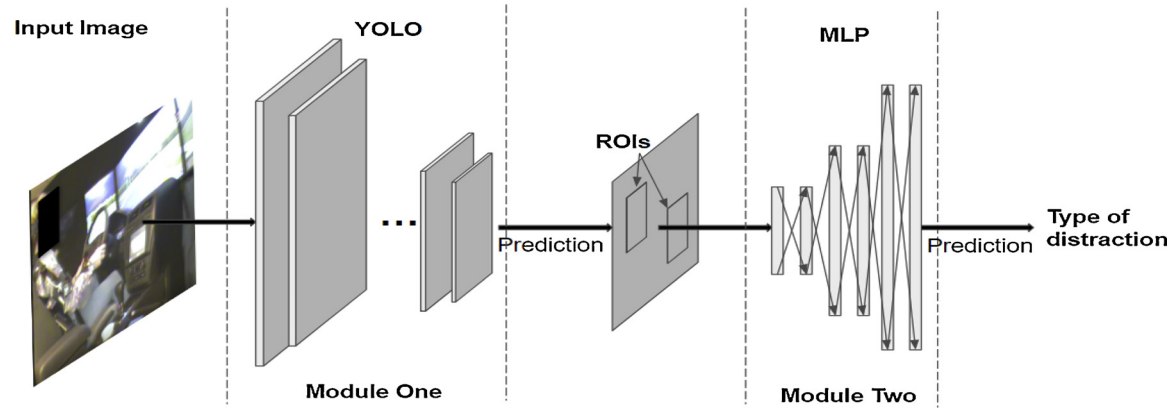

Fig. 1. Method overview. The framework consists of two modules. The first module adopted YOLO for detection of the driver's hand and ear. The second model is a multi-layer perceptron. It predicted the type of distraction based on the spatial patterns of the driver's hand and ear.

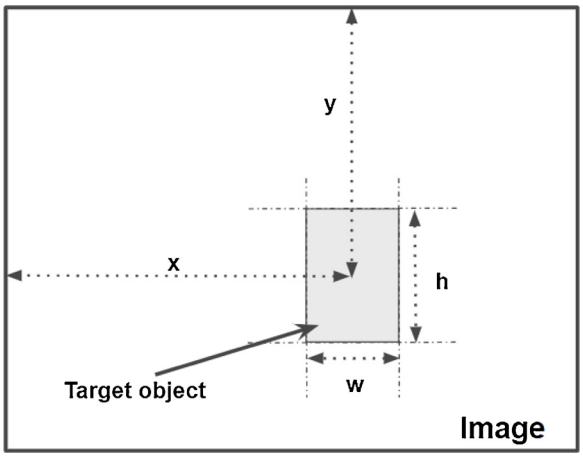

Fig. 2. Formulation of bounding box. x and y represent the coordinates for the center of the bounding box. w and h denote the width and height.

$$\text{BBox} = F_1(\text{Img}) \tag{2}$$

$$\text{BBox} \overset{\text{def}}{=} [x_{\text{hand}}, y_{\text{hand}}, w_{\text{hand}}, h_{\text{hand}}, x_{\text{ear}}, y_{\text{ear}}, w_{\text{ear}}, h_{\text{ear}}] \tag{3}$$

Where F_1 refers to the network in module one; Img represents the input image, which is a N × M × 3 matrix (N stands for number of rows; M denotes number of columns; '3' denotes number of color channels); and BBox is the concatenated vector of the two bounding boxes, which is the output of the first module.

The previously proposed multi-scale faster-RCNN (Ngan Le et al., 2016) is essentially a counterpart of YOLO in module one, which detects the steering wheel and the driver's hands simultaneously, and the intersection of them is the indicator for normal driving. However, due to the projection in the pinhole camera model, the intersection on the image plane does not necessarily mean actual intersection in 3-D space, which could lead to a high miss rate for detecting manual distractions. In this study, the second module seeks to alleviate this problem. The output 8-D vector not only encodes spatial information on the image plane (x and y) but also the depth information through w and h, i.e. objects farther from the camera will have a smaller projected size.

The second module takes the 8-D vector as input and outputs the type of distraction, which can be formulated as follows:

$$C = F_2(\text{BBox}) \tag{4}$$

Where F_2 refers to the network in module two; and C represents the type of distraction (e.g. $C = i$ represents the ith type of distraction). Therefore, the method proposed in this paper can be formulated as:

$$C = F_2(F_1(\text{Img})) \tag{5}$$

In the following sections, the two modules are described in detail.

2.2. Module one: YOLO

YOLO is one of the fastest and most accurate among all of the state-of-the-art object detection algorithms. It was first proposed in Redmon et al. (2016) and was further improved in Redmon and Farhadi (2017), Redmon and Farhadi (2018) with faster speed and more accurate performance. The third version was adopted in the current module one (Fig. 4).

In this module, each input image was first reshaped to 416 × 416 pixels and then divided into $S_h \times S_w$ grid cells (9 × 11 in this study). It was assumed that each grid cell contained one object at most. Second, nine manually defined anchor boxes with different sizes were first centered at the center of each grid cell. A bounding box was then derived by refining the outcomes of the anchor boxes. More specifically, the network first checked the regions covered by anchor boxes, and optimized their size and location until they reached the maximum confidence score. The sizes of anchor boxes were determined initially by running K-nearest neighbors (KNN) on the training set to pick up the most frequent sizes. For example, the bounding boxes covering the right ear over video frames are usually vertical rectangles in the camera view. Thus, the anchor boxes were initialized as a vertical rectangle to accelerate the training. Note that the size of anchor box could be smaller or larger than the size of the grid cell during the initialization and optimization stage, depending on the actual size of the target objects. The following layers then optimized the location as well as the size based on the anchor boxes. The final classification was chosen from the optimized anchor boxes using non-maximum suppression (NMS). B is the number of anchor boxes for each cell (set as nine). So for each image, there were $S_h \times S_w \times B = 891$ anchor boxes in total. The classification flow is illustrated in Fig. 3, . The network minimized the discrepancy between its classification and the ground truth during training, which was measured by the loss function. Details of the loss function can be found in Redmon and Farhadi (2017).

2.3. Module two: a multi-layer perceptron

In this module, the task is to classify an 8-D vector generated from module one. The input 8-D vector encodes the 2D spatial information as well as the depth information of the right hand and ear. The labeler was asked to use their best judgment to guess the location of the ear when the ear was blocked by other body segments. During the training phase, module two would learn to make corresponding classifications on the occluded cases.

To be specific, a six-layer perceptron was designed (see structure in Fig. 5). Batch normalization was added after each dense layer (Ioffe and

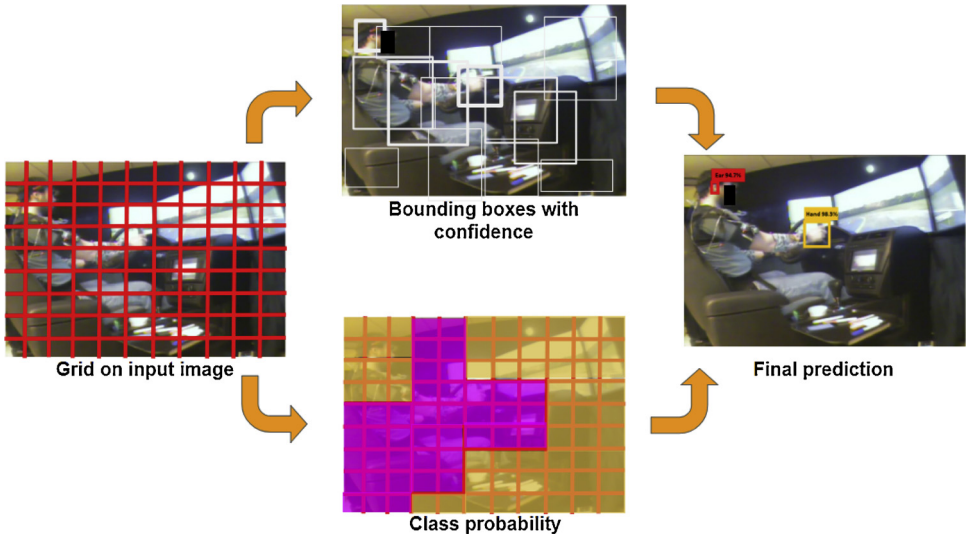

Fig. 3. The YOLO Model. The image is divided into $S_h \times S_w$ grid cells (left image). For each cell, bounding boxes, confidence and class probabilities are predicted. In the bottom picture, the red grid cells have higher probability for the hand, and yellow grid cells are more likely to contain the ear. The upper image shows candidate bounding boxes. Finally, the unified detection framework of these three components finally gives the correct predicted bounding boxes of each objects with class labels (right image). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Szegedy, 2015). The batch normalization layer normalizes the raw output from each dense layer, and mitigates internal covariate shift. It can also accelerate the training. Rectified Linear Units (RELU) (Nair and Hinton, 2010) were chosen as the activation function for each layer. The RELU function filters images with positive responses and avoids gradients saturation by keeping the magnitude of the positive response on the feature map. A dropout layer with dropout probability 0.1 (Srivastava et al., 2014) was added for each layer, which could mitigate overfitting during training. During training, Adam (Kingma and Ba, 2014) was used as the optimizer for updating the parameters. The multi-layer perceptron projects the 8-D vector into a 6-D vector, of which each element represents the confidence score for each type of task, and the cross-entropy of softmax was used as the loss function, which is given as:

$$\hat{z}_i = \frac{\exp(q_i)}{\sum_{i=1}^{N} \exp(q_j)} \tag{6}$$

$$\text{loss}_2 = -\sum_{i=1}^{N} z_i \log \hat{z}_i \tag{7}$$

Where q_i is the confidence score for the ith class, and $\hat{z}_i$ is the softmax of the ith class to avoid numerical issue. z_i is the ground truth one-hot vector. N represents the total number of classes. $loss_2$ is the cross-entropy loss. So the final classification can be written as:

$$C = \arg\max_i \hat{z}_i \tag{8}$$

Learning rate was set to 1.0×10^{-3}. Batch stochastic gradient descent was used to optimize the loss with batch size equals to 150, and the network was trained for 100 epochs.

3. Data collection

3.1. Participants

Twenty participants (8 females and 12 males, 25–55 years old, all with valid drivers' licenses) were recruited for the experiment. This experiment was approved by the New England Institutional Research Board (IRB Study # 1394).

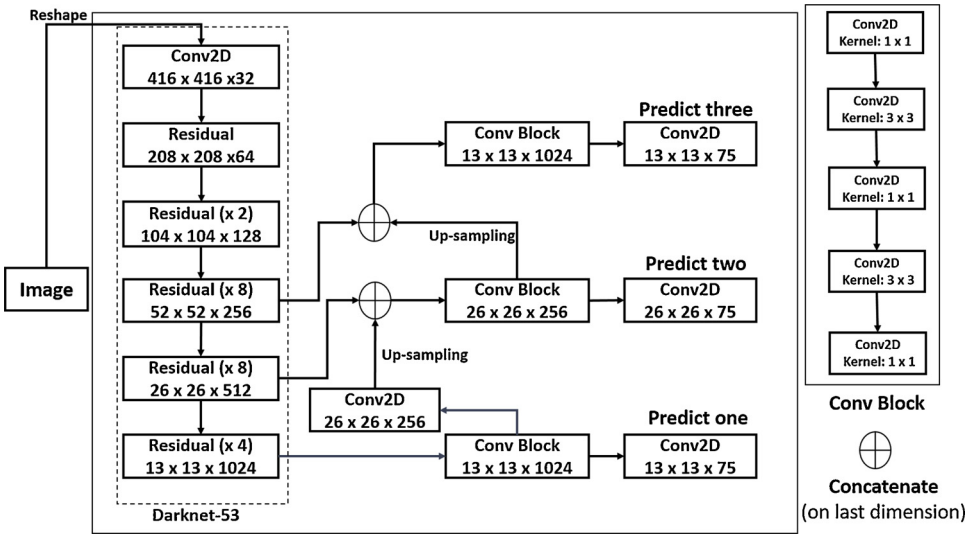

Fig. 4. Structure of YOLO network in module one. $a_1 \times a_2 \times a_3$ represents the dimension of the layer, i.e. width × height × filters. The residual block used 1×1 and 3×3 kernels. All other layers were convolved with 3×3 kernels. The first module takes images as input, and outputs the predicted bounding boxes for target objects.

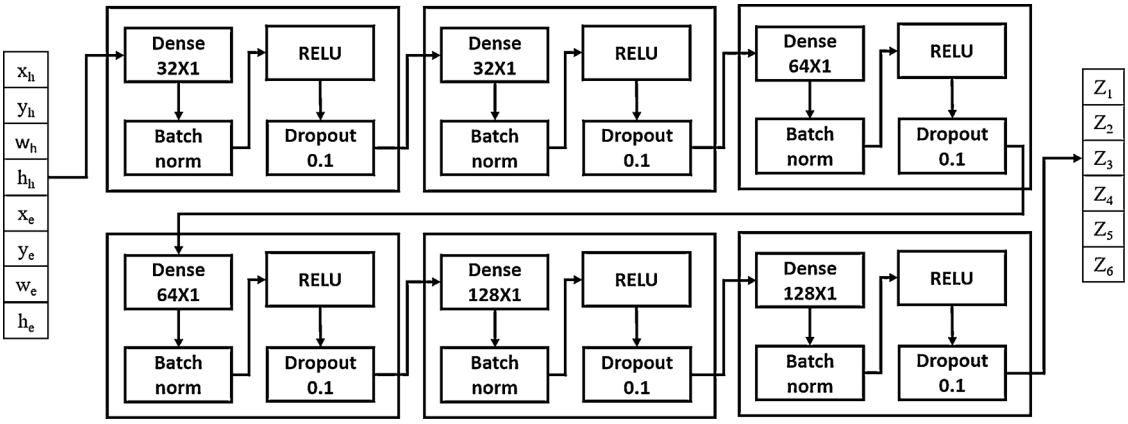

Fig. 5. The structure of multi-layer perceptron in module two. It consists of six dense layers. The first layer takes the bounding boxes as input, and the last layer outputs the type of distraction. (x_h, y_h, w_h, h_h) and (x_e, y_e, w_e, h_e) are the bounding box features of hand and ear, respectively.

3.2. Driving simulator

A RTI driving simulator (Ann Arbor, MI) was used for the study, which is a fixed-based simulator that consists of an open-cab vehicle mock up, including accelerator and brake pedals, steering wheel, dashboard, instrument panel, and center console. Three 46-inch wide screen LCD displays were adopted to present driving environments for the participants, which provided 200° of forward visual angle from the driver's view point. Various driving environments and traffic scenarios were generated using RTI SimCreator and SimVista software. Video footage collected from the right side of the driver was used in this study.

3.3. Driving scene

Participants were driving under various driving situations: in the city street scenario they were (1) driving on a straight road, (2) making a left turn at an intersection, (3) making a right turn at an intersection, (4) stopping and proceeding straight through an intersection with traffic light, and (5) driving on a curved road. Participants also drove in the highway road scenario with both straight road and curved road conditions. During the experiment, violating the traffic rules was not allowed.

3.4. Distraction tasks

During each experiment trial, participants were assigned to five types of non-driving tasks, which include, talking on a cell phone (phone), texting (text), drinking water (drink), using the touchscreen (touchscreen), and placing a marker into the cup holder (marker).

3.5. Procedures

The experiment follows a within-subject design. Before conducting the main experiments, each participant practiced until they felt comfortable to operate the driving simulator. During each experimental block, participants performed the five distracting tasks mentioned above in a random order. Verbal instructions regarding the to-be-performed task were provided through the vehicle speakers. In-vehicle tasks were self-paced and participants pushed a button on the steering wheel when the task was complete. Verbal instructions for the next task were played approximately twenty seconds later. A complete experiment for one participant included six blocks, each of which lasted for approximately fifteen minutes. Five of the six blocks included all the distraction tasks. The rest one did not include any distraction task and was used as the baseline to enhance the performance of classification. Between each block, a five-minute break was given. Due to simulator sickness, not every participant completed all six blocks. Across all participants, data from 97 blocks were available for modeling.

3.6. Data processing

One experimental block was randomly selected from each participant. Considering that each video had more than 27,000 frames, only one frame was extracted from every five frames to reduce redundancy. There were 106,677 frames extracted from twenty participants in total. Fig. 6 shows examples of performing different distracting tasks. Each task can be represented with four key events, which are 'start', 'initiate', 'return' and 'end'. 'Start' represents the frame the participant starts move their right hand off the wheel. 'Initiate' represents the frame when the participant initiates contact with the target object. 'Return' denotes the frame when the task is finished. 'End' represents the right hand moves back to the steering wheel. The frames between 'start' and 'end' were annotated with the corresponding distraction class. Then the driver's right hand and right ear were labeled in each frame using MATLAB computer vision toolbox. Fig. 7 plots the hand locations of four participants during different distracting tasks. Clustering can be clearly seen in the picture. 5626 frames were extracted from each participant on average (Table 1). The extracted frames were used for training and testing, the details of which will be given in the following sections.

4. Result

4.1. Model training

The two above mentioned modules were trained separately. For the first module, the model was pre-trained by the hand annotations from five participants plus VIVA hand tracking dataset (a public hand detection dataset) (Viva hand tracking dataset, 2016), which include 23,705 and 4744 samples, respectively. The hand and ear annotations were then used for training on the pre-trained model, which includes 1000 samples in total (100 from ten participants). Because VIVA hand tracking dataset includes diverse hand annotations in naturalistic settings, adding this dataset in the pre-training step can save the number of training samples while still keeping the capacity of the network. Including this dataset can also enhance the generalizing ability of the model and thus mitigate overfitting. For the second module, the annotated hand and ear and activity IDs were used as paired data for training, which included 1000 samples in total (100 from ten participants, same with the second training step in module one).

In summary, labeled data from ten participants were used for training, and the rest data from the other ten participants were used for testing. During training, module one took three hours, and module two took less than one minute. The hardware information is summarized in

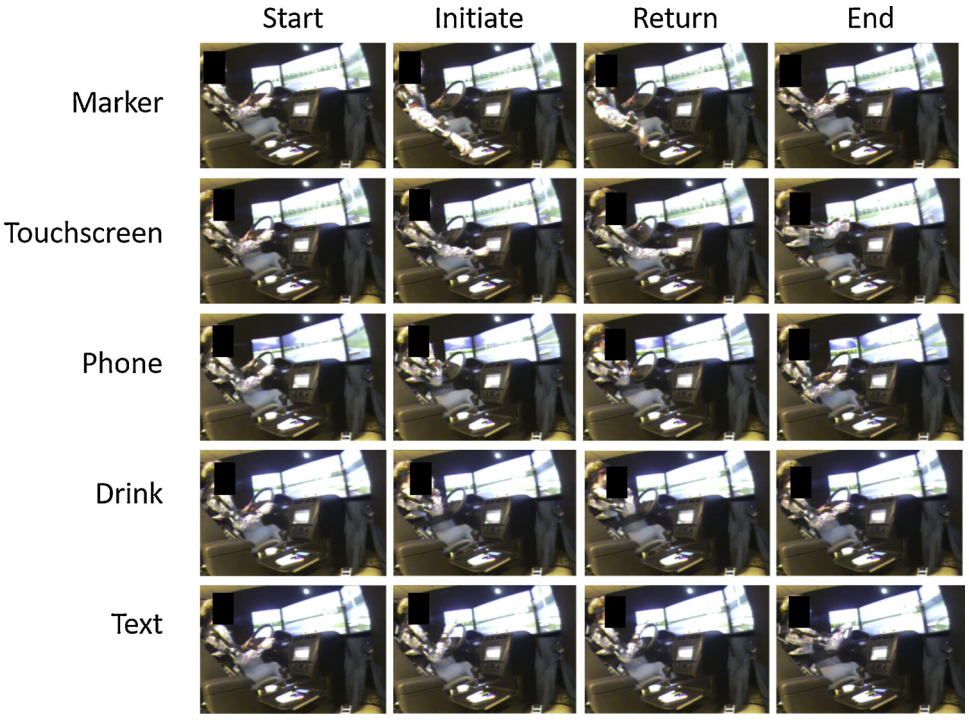

Fig. 6. Distracting tasks. Each task can be described by four key events, which are 'start', 'initiate', 'return' and 'end'. Note that 'phone' only refers to talking on the phone, which is different from 'text'.

Table 2.

4.2. Evaluation metrics

Since the two modules were trained separately, they were evaluated with different metrics. The first module was evaluated with mean accuracy precision (mAP) Everingham et al. (2010).

- **Mean average precision (mAP):** mAP calculates the mean of the average precision rate among different target object classes (i.e. hand and ear in module one) and each detection of that class. It measures the overlapping area of the bounding boxes from the ground truth and detection result, and was used to evaluate the detection accuracy of module one. Specifically:

$$\text{IoU} = \frac{\text{Intersection of BBox}_{\text{pred}} \text{ and BBox}_{\text{gt}}}{\text{Union of BBox}_{\text{pred}} \text{ and BBox}_{\text{gt}}} \tag{9}$$

$$\text{mAP} = \frac{\text{Number of IoUs greater than 0.5}}{\text{Total number of IoUs}} \tag{10}$$

Where $BBox_{pred}$ and $BBox_{gt}$ represent the bounding box of the classification and ground truth, respectively; Intersection over Union (*IoU*) calculates the intersection of two bounding boxes divided by their union; and *mAP* is the percentage of classifications having IoU greater than 0.5.

The second module was evaluated with classification metrics. Common classification metrics include precision, recall and F1-score, which are

Fig. 7. Right hand tracking. The location, color and size of the points are encoded by the x & y, w & h, and type of distraction, respectively. Note that (x, y) represents the center point of the bounding box (illustrated in Fig. 2).

Table 1
Number of frames extracted from each participant. One block was randomly selected from each participant. One frame was extracted from every five frames to avoid redundancy.

ID	Total	ID	Total	ID	Total	ID	Total
Participant 01	6945	Participant 06	5190	Participant 11	5451	Participant 16	5341
Participant 02	6698	Participant 07	5359	Participant 12	5035	Participant 17	6009
Participant 03	5150	Participant 08	5235	Participant 13	5828	Participant 18	5844
Participant 04	4912	Participant 09	5227	Participant 14	5778	Participant 19	5930
Participant 05	5828	Participant 10	5445	Participant 15	6096	Participant 20	5220

Table 2
Hardware information.

Compiler	Python 2.7
Backend	Keras with tensorflow
CPU	Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60 GHz
GPU	2 × Titan V

given as follows:

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{11}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{12}$$

$$F1\text{score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

Where *tp* represents true positives; *fn* is the number of false negatives; and *fp* is the false positives. However, they are designed for binary classification and tend to over-estimate the error rate for multi-class classification. For a more comprehensive evaluation, metrics proposed in Sokolova and Lapalme (2009) for multi-class classification were adopted.

- **Average accuracy and error rate:** The average accuracy and error rate measure the overall performance of the multi-class classification (Van Asch, 2013), and are calculated as follows:

$$\text{Accuracy}_{\text{average}} = \frac{\sum_{i=1}^{l} \frac{\text{tp}_i + \text{tn}_i}{\text{tp}_i + \text{fn}_i + \text{fp}_i + \text{tn}_i}}{l} \tag{14}$$

$$\text{Error}_{\text{average}} = \frac{\sum_{i=1}^{l} \frac{\text{fp}_i + \text{fn}_i}{\text{tp}_i + \text{fn}_i + \text{fp}_i + \text{tn}_i}}{l} \tag{15}$$

  Where tp_i represents true positives for the ith class; tn_i denotes true negatives; fn_i is the number of false negatives; and fp_i is the number of false positives. l equals total number of classes.
- **Micro-averaging:** The micro-averaging measures give each class a weight according to the sample size (Van Asch, 2013). Therefore, they favor the classes with more samples. They are given as follows:

$$\text{Precision}_{\mu} = \frac{\sum_{i=1}^{l} \text{tp}_i}{\sum_{i=1}^{l} (\text{tp}_i + \text{fp}_i)} \tag{16}$$

$$\text{Recall}_{\mu} = \frac{\sum_{i=1}^{l} \text{tp}_i}{\sum_{i=1}^{l} (\text{tp}_i + \text{fn}_i)} \tag{17}$$

$$F1\text{score}_{\mu} = \frac{2 \cdot \text{Precision}_{\mu} \cdot \text{Recall}_{\mu}}{\text{Precision}_{\mu} + \text{Recall}_{\mu}} \tag{18}$$

  Where $precision_{\mu}$ and $Recall_{\mu}$ evaluate the effectiveness of a classifier to identify class labels if calculated from the sums of per-text decisions, and the $Fscore_{\mu}$ reveals the relationships between the data's positive labels and those from the classifier based on the sums of per-text decisions.
- **Macro-averaging:** The macro-averaging treats all classes more equally, which is more suitable for an imbalanced dataset. The formulas are given in the following:

$$\text{Precision}_{M} = \frac{\sum_{i=1}^{l} \frac{\text{tp}_i}{\text{tp}_i + \text{fp}_i}}{l} \tag{19}$$

$$\text{Recall}_{M} = \frac{\sum_{i=1}^{l} \frac{\text{tp}_i}{\text{tp}_i + \text{fn}_i}}{l} \tag{20}$$

$$F1\text{score}_{M} = \frac{2 \cdot \text{Precision}_{M} \cdot \text{Recall}_{M}}{\text{Precision}_{M} + \text{Recall}_{M}} \tag{21}$$

  The macro-averaging measures the agreement of the average per-class of a classifier to identify class labels. These systematic measures gave a more objective and reliable evaluation for the algorithm.

4.3. Model validation

The whole framework ran at 28 FPS. For module one, the mAP for hand and ear detection was 56.6%. Besides multi-class classification, the proposed network was also evaluated with binary classification. That is, five distracting tasks were all labeled by single category of “distracted driving”, and the network predicted whether a distracting activity is present for each frame using the same structure for the six-class classification. The average of k-fold ($k = 5$) cross-validation was taken as the final result.

Table 3 shows the confusion matrix for multi-class classification. Let V_{ij} represent the value of the ith row and the jth column in the confusion matrix. V_{ij} equals to the number of the ith distraction that were classified as the jth distraction.

Table 4 and Table 5 show the evaluation results and multi-class results under two averaging metrics introduced in the previous section. The algorithm achieved results in discerning distracting activities and normal driving with F1-score 0.74 and 0.87, respectively, but also revealed variation in accuracy of detecting different types of distraction. Normal driving had the highest F1-score (0.84), followed by phone (0.82), and touchscreen (0.69). The F1-scores for other types of distraction were below 0.5. For multi-class classification metrics, the algorithm achieved results in terms of average accuracy (0.92). For the micro-averaging, the algorithm achieved an average F1-score of 0.75, and 0.60 for the macro-averaging. The algorithm demonstrated better performance under micro-averaging because the samples of normal driving have much more weight than other classes under micro-averaging compared to macro-averaging.

The normal driving resulted in the most type I and type II error (around 71%) among all distracting activities. This is because the driver's right hand is very close to the wheel at the ‘start’ and ‘end’ of a distracting task (Fig. 6), and can thus be confused with normal driving.

Table 3
Confusion matrix for six-class classification. Rows represent the actual class and columns represent the predicted class. The diagonal entries show the number of samples correctly classified.

	Normal driving	Marker	Drink	Touchscreen	Text	Phone	Total
Normal driving	**31,108**	801	1185	2046	1272	596	37,008
Marker	1255	**820**	71	130	96	77	2449
Drink	806	184	**941**	128	269	266	2594
Touchscreen	320	0	15	**3044**	2	1	3382
Text	3003	123	175	312	**2390**	184	6187
Phone	386	40	57	70	134	**4225**	4912
Total	36,878	1968	2444	5730	4163	5349	56,532

Table 6 shows the similarity level of each pair of distractions. The value on the *i*th row and the *j*th column denotes the percentage of the *j*th distraction in the type I error cases of the *i*th distraction. Drink-touchscreen, touchscreen-marker, and touchscreen-text are the top three pairs that were the most frequently misclassified by the algorithm. The reason is that using the touchscreen event is the most 'neutral' among all events. It lies in between the hand lifting up and reaching down, and it involves the minimum manual moving distance from the wheel. Therefore, the spatial features of other classes were frequently misclassified as touchscreen.

5. Discussion

The primary aim of this study was to develop a robust and efficient image-based algorithm for detecting driving distraction. Table 7 compares the overall performance of the proposed method with other similar studies intended for driving distraction-related classification (note that the overall accuracy in each study should be interpreted with caution since the algorithms were evaluated on different datasets and hardware). Among these algorithms, the neural network-based algorithms (MS-RCNN (Ngan Le et al., 2016) and LSTM (Wollmer et al., 2011)) demonstrated higher overall accuracy, while the traditional machine learning algorithms were better in efficiency (Adaboost, Random Forest (Seshadri et al., 2015), SVM (Li et al., 2017), and RF model (Atiquzzaman et al., 2018)). Our method demonstrated comparable accuracy in both binary and multi-class classification while achieving the top efficiency through leveraging the CNN and multi-layer perceptron. The convolutional operations and parallel computation abilities of GPUs made the CNN an efficient low-level feature extractor. The multi-layer perceptron was easy to implement and demonstrated great performance in regression over low-dimensional data. Moreover, stacking the two neural network-based modules and loading them onto GPUs as a whole further enhanced the network efficiency.

The architecture of the proposed method also provides flexibility for future modification and improvements. The first module does not need further fine-tuning if improvements are later made to the second module, because the accuracy of the second module only depends on its structure and the output of the first module. For example, more advanced structures can be stacked into the second module with the first module unchanged. The categories of distractions could also be extended towards a more comprehensive monitoring system through the adjustment of module two. In comparison, an end-to-end pipeline may lose the flexibility.

However, several limitations are faced in this study. The results revealed substantial variation in F1-scores among different classes. The reason can be two-fold. First, the proposed algorithm is detection-

Table 5
Multi-class classification measures (Sokolova and Lapalme, 2009). μ and M indices represent micro- and macro-averaging.

Measure	Value
Average accuracy↑	0.92
Error rate↓	0.08
Precision_{μ}↑	0.75
Recall_{μ}↑	0.75
Fscore_{μ}↑	0.75
Precision_{M}↑	0.59
Recall_{M}↑	0.61
Fscore_{M}↑	0.60

Table 6
Similarity matrix. The entry at (*i*, *j*) represents the percentage of the *j*th distraction in the false-positives of the *i*th distraction, e.g. for the false-positives of marker event, drink event took up 19% of them.

	Marker	Drink	Touchscreen	Text	Phone
Marker	–	0.19	0.35	0.26	0.21
Drink	0.22	–	0.15	0.32	0.31
Touchscreen	0	0.83	–	0.11	0.06
Text	0.15	0.22	0.39	–	0.23
Phone	0.13	0.19	0.23	0.45	–

Table 4
Precision, recall and F1-score of testing result. ↑ denotes a larger value is preferred; ↓ means a smaller value is preferred.

	Class	Precision↑	Recall↑	F1-score↑	Type I error↓	Type II error↓
Binary classification	Normal driving	0.85	0.90	0.87	0.15	0.10
	Distracted driving	0.78	0.70	0.74	0.22	0.30
	Average	0.82	0.80	0.81	0.19	0.20
Multi-class classification	Normal driving	0.84	0.84	0.84	0.16	0.16
	Marker	0.42	0.34	0.37	0.58	0.66
	Drink	0.38	0.36	0.37	0.62	0.64
	Touchscreen	0.53	0.90	0.69	0.47	0.10
	Text	0.57	0.39	0.46	0.43	0.61
	Phone	0.79	0.86	0.82	0.21	0.14
	Average	0.59	0.62	0.59	0.41	0.39

Table 7
Comparison of distraction-related classification algorithms. To be fair, the average accuracy of the proposed method was calculated as the average of the precision in Table 4, since other methods did not include multi-class classification metrics. ↑ denotes a larger value is preferred; ↓ means a smaller value is preferred. The best results are bolded. MS-RCNN stands for multi-scale regional convolutional neural network. RF model is one kind of tree-based algorithm. LSTM denotes long short-term memory, one kind of the recurrent neural network.

	Method	Overall accuracy↑	FPS↑	Classes
Binary classification	Proposed method	0.82	**28**	2
	MS-RCNN (Ngan Le et al., 2016)	0.85	0.09	2
	Adaboost (Seshadri et al., 2015)	0.84	7.5	2
	SVM (Li et al., 2017)	0.95	-	2
	Random Forest (Seshadri et al., 2015)	0.80	7.5	2
	RF model (Atiquzzaman et al., 2018)	0.86	-	2
	LSTM (Wollmer et al., 2011)	**0.96**	2	2
Multi-class classification	Proposed method	**0.59**	**28**	6
	LSTM (Wollmer et al., 2011)	0.54	2	3
	LSTM (Wollmer et al., 2011)	0.43	2	6

based, while the human movements are continuous. When the activity is at its 'start' and 'end', the features are very close to the previous activity and the algorithm could misinterpret the action. Thus, the performance of the detection-based algorithm could be less robust at making distinction among continuous events. Second, although extracting spatial features of the hand and ear is sufficient for many distractions, specific distractions may need more features like gestures or other parts of body for their encoding. Regarding the efficiency, the algorithm was implemented at 28 FPS, which is nearly fast enough for real-time monitoring. The bottleneck of efficiency is mainly from the YOLO module. To further improve the speed, accuracy would need to be sacrificed (e.g. the tiny YOLO, a simplified version of YOLO, and it reportedly can run at 150 FPS on Titan while sacrificing mAP by around 24% (Redmon and Farhadi, 2017)). The cost of the training step could be another limitation for the proposed method. The generalizing abilities of the algorithm largely depend on the training dataset due to the data-driven nature of deep learning. The labeling for the training set could take weeks and proficient labelers are needed.

Future work will seek to address the limitations in this study. First, more advanced network architectures could be explored to further increase the robustness of the algorithm, like building a hierarchical structure for each activity and combining the temporal information, so that each activity can be further divided into more sub-tasks with more discernible features. Future studies could also use the recurrent neural network (RNN) to learn the temporal features, which may further improve the robustness of the algorithm.

Second, more features in driving cab could be extracted to improve the inference accuracy. Ideally, the features encoded in the collected data should be the sufficient statistics of a driver's status to guarantee the best accuracy. As drivers' behaviors under naturalistic settings vary substantially, the more data that are collected, the more likely the data include the sufficient statistics. Therefore, other data, like drivers' facial features, heart rate and gesture, could be collected for training a unified and more robust neural network. However, labeling these additional data could also require considerably more labor.

6. Conclusion

In this paper, a novel algorithm for detecting drivers' manual distraction is proposed. The algorithm consists of two modules, where the first module detects the driver's right hand and right ear, and the second module predicts the status of drivers based on their spatial patterns. The algorithm was evaluated with driving videos collected from a driving simulator. Results indicated that the proposed algorithm is comparable with results from similar work in overall accuracy, and is more efficient than other methods. Future work will focus on exploring data fusion and temporal relations of frames to further enhance the robustness of driving distraction detection.

Author contribution

Li Li: Conceptualization, Methodology, Software, Formal analysis, Writing-Original Daft.

Boxuan Zhong: Methodology, Software, Writing-Original Daft.

Clayton Hutmacher Jr.: Data Curation, Writing-Review & Editing.

Yulan Liang: Conceptualization, Investigation.

William J. Horrey: Conceptualization, Investigation, Writing-Review & Editing.

Xu Xu: Investigation, Data Curation, Conceptualization, Funding acquisition, Writing-Review & Editing.

Conflict of Interest

All authors declare that there is no proprietary, financial, professional or other personal interest of any nature or kind in any product, service or company that could be construed as influencing the position presented in the manuscript.

Acknowledgements

This manuscript is based upon work supported by the National Science Foundation under Grant No. 1822477. The authors are also grateful to Jacob Banks, Niall O'Brien, Amanda Rivard, Luci Simmons, and Sarah Hong for assistance in data collection.

References

Atiquzzaman, M., Qi, Y., Fries, R., 2018. Real-time detection of drivers' texting and eating behavior based on vehicle dynamics. Transp. Res. Part F: Traffic Psychol. Behav. 58, 594–604.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Fernández, A., Usamentiaga, R., Carús, J., Casado, R., 2016. Driver distraction using visual-based sensors and algorithms. Sensors 16 (11), 1805.

Foss, R.D., Goodwin, A.H., 2014. Distracted driver behaviors and distracting conditions among adolescent drivers: findings from a naturalistic driving study. J. Adolesc. Health 54 (5), S50–S60.

Fridman, L., 2018. Human-Centered Autonomous Vehicle Systems: Principles of Effective Shared Autonomy. arXiv preprint arXiv:1810.01835.

Gallahan, S.L., Golzar, G.F., Jain, A.P., Samay, A.E., Trerotola, T.J., Weisskopf, J.G., Nathan, L., 2013. Detecting and mitigating driver distraction with motion capture technology: distracted driving warning system. In: 2013 IEEE Systems and Information Engineering Design Symposium. IEEE. pp. 76–81.

Gokturk, S.B., Bouguet, J.-Y., Tomasi, C., Girod, B., 2002. Model-based face tracking for view-independent facial expression recognition. In: Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition. IEEE. pp. 287–293.

Hurtado, S., Chiasson, S., 2016. An eye-tracking evaluation of driver distraction and unfamiliar road signs. In: Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. ACM. pp. 153–160.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv:1502.03167.

Kim, J., Kim, K., Yoon, D., Koo, Y., Han, W., 2016. Fusion of driver-information based

driver status recognition for co-pilot system. In: 2016 IEEE Intelligent Vehicles Symposium (IV). IEEE. pp. 1398–1403.
Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
Li, Z., Bao, S., Kolmanovsky, I.V., Yin, X., 2017. Visual-manual distraction detection using driving performance indicators with naturalistic driving data. IEEE Trans. Intell. Transp. Syst. 19 (8), 2528–2535.
Li, L., Xie, Z., Xu, X., 2019. Recognition of manual driving distraction through deep-learning and wearable sensing. Proceedings of the International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, vol. 2019 22–28.
Liang, Y., Reyes, M.L., Lee, J.D., 2007. Real-time detection of driver cognitive distraction using support vector machines. IEEE Trans. Intell. Transp. Syst. 8 (2), 340–350.
Majumder, A., Behera, L., Subramanian, V.K., 2016. Automatic facial expression recognition system using deep network-based data fusion. IEEE Trans. Cybern. 48 (1), 103–114.
Miyaji, M., Kawanaka, H., Oguri, K., 2009. Driver's cognitive distraction detection using physiological features by the adaboost. In: 12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC'09. IEEE. pp. 1–6.
Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10) 807–814.
Ngan Le, T.H., Zheng, Y., Zhu, C., Luu, K., Savvides, M., 2016. Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 46–53.
Ranney, T.A., 2008. Driver Distraction: A Review of the Current State-of-Knowledge. Technical Report.
Redmon, J., Farhadi, A., 2017. Yolo9000: Better, Faster, Stronger. arXiv preprint.
Redmon, J., Farhadi, A., 2018. Yolov3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 779–788.
Schwarz, C., Gaspar, J., Miller, T., Yousefian, R., 2019. The detection of drowsiness using a driver monitoring system. Traffic Injury Prev. 20 (Suppl. 1), S157–S161.
Seshadri, K., Juefei-Xu, F., Pal, D.K., Savvides, M., Thor, C.P., 2015. Driver cell phone usage detection on strategic highway research program (shrp2) face view videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 35–43.
Sigari, M.H., 2009. Driver hypo-vigilance detection based on eyelid behavior. In: Seventh International Conference on Advances in Pattern Recognition, 2009. ICAPR'09. IEEE. pp. 426–429.
Sigari, M.-H., Fathy, M., Soryani, M., 2013. A driver face monitoring system for fatigue and distraction detection. Int. J. Veh. Technol. 2013.
Sigari, M.-H., Pourshahabi, M.-R., Soryani, M., Fathy, M., 2014. A Review on Driver Face Monitoring Systems for Fatigue and Distraction Detection.
Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45 (4), 427–437.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Stutts, J., Feaganes, J., Reinfurt, D., Rodgman, E., Hamlett, C., Gish, K., Staplin, L., 2005. Driver's exposure to distractions in their natural driving environment. Accid. Anal. Prev. 37 (6), 1093–1101.
Van Asch, V., 2013. Macro-and Micro-Averaged Evaluation Measures [Basic Draft]. CLiPS, Belgium, pp. 1–27.
Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W., Levi, D., 2015. Driver gaze tracking and eyes off the road detection system. IEEE Trans. Intell. Transp. Syst. 16 (4), 2014–2027.
Viva hand tracking dataset. http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-tracking/ (accessed 2016).
Wollmer, M., Blaschke, C., Schindl, T., Schuller, B., Farber, B., Mayer, S., Trefflich, B., 2011. Online driver distraction detection using long short-term memory. IEEE Trans. Intell. Transp. Syst. 12 (2), 574–582.
Zhang, K., Huang, Y., Du, Y., Wang, L., 2017. Facial expression recognition based on deep evolutional spatial-temporal networks. IEEE Trans. Image Process. 26 (9), 4193–4203.