Driver state estimation by convolutional neural network using multimodal sensor data

Sejoon Lim✉ and Ji Hyun Yang

A driver state estimation algorithm that uses multimodal vehicular and physiological sensor data is proposed. Deep learning is applied to the fused multimodal data rather than each modality being treated as a different feature. A convolutional neural network model is developed and the driver state estimation algorithm is implemented using Google TensorFlow. The results show that deep learning is a very promising approach for driver state estimation compared with previously studied algorithms, such as dynamic Bayesian networks.

Introduction: Approximately 94% of all car accidents are caused by driver error and 75% of this total is due to recognition and decision errors [1]. The identification of abnormal driver states, such as drowsiness, distraction, and high workload, is essential for preventing human-error-related car accidents. State-of-the-art sensor technologies enable the measurement of vehicle- and driver-related signals. With the development of sensor technology and on-board computational units, systems for identifying and warning drivers (or even controlling the cars themselves) are becoming essential features in next-generation vehicles. One of the key technical issues that must be solved in the development of autonomous vehicles is human–machine interaction, which includes driver state estimation and countermeasures.

In this Letter, we estimated the states of driver drowsiness, visual distraction, cognitive distraction, and high workload of multiple subjects based on sensory data collected from a driving simulator.

We then developed a driver state detection algorithm based on deep learning and demonstrated its performance. Deep learning has been successfully applied in many areas, including computer vision, speech recognition, and multimodal data fusion [2, 3]. We use both driver physiological data and vehicle data collected in the simulated driving environment. Our approach was to use the deep learning algorithm, which has rarely been used in driver state estimation using multimodal data to address the driver state detection problem. Rather than treating types of data differently, we fused the multimodal data time sequence using a two-dimensional matrix, wherein one dimension is used for different sensor data and the other is used for time.

We examined the correct detection rate (CDR) and false alarm rate (FAR) for our driver state estimation problems, and then compared them with the performance of a previous study that employed the dynamic Bayesian network model [4].

Vehicle and physiology data: Study participants included 35 subjects in their 20 and 30 s from whom we collected vehicle and human data for the four abnormal driver states and one normal state. The subjects were given driving and non-driving tasks in a simulated driving environment, as shown in Fig. 1. We obtained drowsiness data from driving simulator experiments in which the subjects were asked to sleep no more than 4 h the night before the experiment. We obtained visual distraction data by giving the subjects secondary tasks such as mobile phone activities while they were driving. We obtained cognitive distraction data by giving the subjects tasks such as speaker-phone mode conversations with the experimenter. We obtained high workload data by having the subjects drive in a stressful environment in which the vehicle in front of the driver's car made frequent sudden stops after the subjects had been asked to drive at a speed of 40 km/h. We assumed that abnormal states occurred during all study sections in which a task had been assigned. Details of the experimental set-up and apparatus can be found in [4, 5].

Vehicle, vision, voice, and physiological information were collected by relevant sensor systems installed in the driving simulator environment. The sampling frequency of the data was 30 Hz. Vehicle information included the vehicle's velocity, longitudinal acceleration, lateral acceleration, steering wheel angle, and gas pedal angle. Vision information included the participants' blinking rate, percentage of eye closure, and facial direction. Voice information included the participants' audio amplitudes. Physiological information included the participants' heart rate, respiration rate, galvanic skin response, and body temperature.

We separated the data into training, validation, and test sets. We used the training and validation data sets to train our model, and used the test data set to test the performance of our trained model.

Convolutional neural network model: The objective of this Letter was to determine driver state as falling into one of the following categories: drowsiness, visual distraction, cognitive distraction, high workload, or normal.

Fig. 1 *Driving simulator*

We used convolutional neural network (CNN) with rectified linear units (ReLUs), max pooling, dropout, and softmax regression. CNNs are feed-forward neural networks designed to deal with large input spaces, such as those seen in image classification tasks. CNNs are constructed by alternatively stacking convolutional and pooling layers.

The structure of our CNN is shown in Fig. 2. We based our CNN on our intuition that there should be some local pattern in time and between the multimodal sensor data. This pattern primitively captures a signal when the driver state is abnormal. Using the CNN approach, we can prevent overfitting of the data as well as reduce the computational cost.

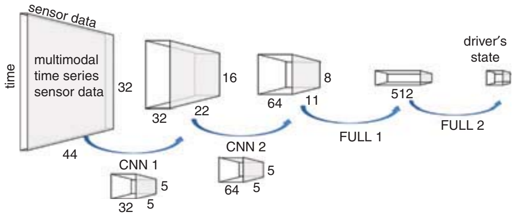

Fig. 2 *CNN model*

We used a 32 × 44 matrix as the input for our CNN model, where 32 represents the number of samples in 0.1 s time slots used, and 44 is the number of data collected from the four kinds of sensors used in this study. We used two convolution layers followed by max pooling layers. Convolutional layer 1 has 5 × 5 filters and a depth of 32, and convolutional layer 2 has 5 × 5 filters and a depth of 64. We used ReLU as an activation function for each neuron in the convolutional layers. Then, we connected the output of layer 2 to a fully connected layer with a depth of 512, followed by a 0.5 dropout rate. Finally, we used softmax regression to obtain the desired output: the probability of the driver being in a normal state or one of the abnormal states of drowsiness, visual distraction, cognitive distraction, and high workload.

Result: Using the driving simulator environment equipped with multimodal sensor systems, we collected about 4 h of driving data for each state: drowsiness, visual distraction, cognitive distraction, high workload, and normal. We divided the data into training and test data sets, and chunked the data into 32 × 44 matrices, to be fed into our CNN model. We labelled each input according to the driver's state when the data was collected.

We used Google's TensorFlow Application Programming Interface to implement our CNN model [6], and trained the model on a server computer with two 14-core 2.4 GHz CPUs and 64 GB of RAM. We see in Fig. 3 that the loss function decreases as the number of training iterations increases. We computed the loss function as a cross-entropy over the softmax regression output of the final full layer of the CNN model, plus a regularisation term. Fig. 4 shows the trained filter outputs for CNN layers 1 and 2, in which the visual patterns in the multimodal sensor data are identified. The time taken to train the data was about 100 min. After training the CNN with the training data, we tested the performance using the test data on a laptop computer with a quad-core 2.2 GHz CPUs and 16 GB of RAM.