

Visualization of Driving Behavior Based on Hidden Feature Extraction by Using Deep Learning

HaiLong Liu, *Student Member, IEEE*, Tadahiro Taniguchi, *Member, IEEE*, Yusuke Tanaka, Kazuhito Takenaka, *Member, IEEE*, and Takashi Bando, *Member, IEEE*

Abstract—In this paper, we propose a visualization method for driving behavior that helps people to recognize distinctive driving behavior patterns in continuous driving behavior data. Driving behavior can be measured using various types of sensors connected to a control area network. The measured multi-dimensional time series data are called driving behavior data. In many cases, each dimension of the time series data is not independent of each other in a statistical sense. For example, accelerator opening rate and longitudinal acceleration are mutually dependent. We hypothesize that only a small number of hidden features that are essential for driving behavior are generating the multivariate driving behavior data. Thus, extracting essential hidden features from measured redundant driving behavior data is a problem to be solved to develop an effective visualization method for driving behavior. In this paper, we propose using deep sparse autoencoder (DSAE) to extract hidden features for visualization of driving behavior. Based on the DSAE, we propose a visualization method called a *driving color map* by mapping the extracted 3-D hidden feature to the red green blue (RGB) color space. A driving color map is produced by placing the colors in the corresponding positions on the map. The subjective experiment shows that feature extraction method based on the DSAE is effective for visualization. In addition, its performance is also evaluated numerically by using pattern recognition method. We also provide examples of applications that use driving color maps in practical problems. In summary, it is shown the driving color map based on DSAE facilitates better visualization of driving behavior.

Index Terms—Data visualization, deep learning, driving behavior analysis, feature extraction.

I. INTRODUCTION

VISUALIZATION of driving behavior can help drivers to review and understand their driving behavior. Effective review of driving behavior can contribute to improving their

driving behavior and promoting safe driving. Besides, efficient visualization of driving behavior enables users, e.g., driving school teachers, security officers at transportation companies, and accident investigators, to monitor and investigate driving behavior of cars intuitively.

Visualization of driving behavior could be useful when a person needs to review his/her driving behavior because a user who wants to review their driving behaviors cannot intuitively understand their behavior by watching driving behavior data, i.e., raw time series data. In contrast, reviewing recorded video takes a long time and is not so efficient to find interesting moments in his/her trip. The effective visualization of driving behavior can help drivers to understand their good and bad driving habits. Meanwhile, to ensure safe driving, providing information about previous driving behaviors after driving to users, such as driving school teachers, security officers at transportation companies, and accident investigators is also important. For example, a driving school teacher could understand the driving behavior of students intuitively by visualizing the students' driving behavior. In addition, security officers at a transportation company could monitor the driving behavior of a taxi or bus driver remotely by visualization, and thus, they may inform a driver promptly when a dangerous driving behavior is visualized. Moreover, we expect that the visualization of driving behavior would be useful for accident investigations. The visualization of driving behavior can allow investigators to determine whether unusual driving behavior had led to an accident. Thus, an effective visualization method of driving behavior will be useful for developing various support systems for monitoring, reviewing and analyzing driving behavior, and driving assistance systems.

Many advanced driving assistance systems (ADASs) have been focusing on assisting a driving behavior directly, i.e., vehicle assistance control [1]–[3]. They even participate in controlling the vehicle directly when the car confronts a danger [4], [5]. Driving environment detection [6], driving behavior recognition [7], prediction [8], selective darkening of the windshield [9] and determining utterance timing of driving agents [10] are also examples of ADASs. Although the visualization of driving behavior has been gathering less attention in the fields of intelligent vehicles than ADASs, the visualization of driving behavior is an extremely important issue. In this study, we propose a method that can visualize driving behavior effectively. It is shown that the method can help people to recognize distinct patterns in driving behavior.

Manuscript received May 8, 2016; revised September 17, 2016 and December 12, 2016; accepted December 31, 2016. Date of publication February 2, 2017; date of current version August 28, 2017. The Associate Editor for this paper was Z. Duric.

H. Liu is with the Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan, and with the Japan Society for the Promotion of Science (e-mail: liu@em.ci.ritsumei.ac.jp).

T. Taniguchi is with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan (e-mail: taniguchi@em.ci.ritsumei.ac.jp).

Y. Tanaka and K. Takenaka are with the Corporate Research and Development Division1, Sensing System Research and Development Department, DENSO Corporation, Aichi 448-8661, Japan (e-mail: yuusuke_tanaka@denso.co.jp; kazuhito_takenaka@denso.co.jp).

T. Bando is with DENSO International America, Inc., San Jose, CA 95110 USA (e-mail: takashi_bando@denso-diam.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2017.2649541

Driving behavior can be measured and recorded by various types of sensor via a control area network (CAN), e.g., the accelerator opening rate, brake master-cylinder pressure, and steering angle. We regard driving behavior data as multi-dimensional time-series data that is formed by assembling various types of sensor information. Each type of sensor information comprises a dimension of the driving behavior data. Here, the important property of the driving behavior data is that each time series data observed by each sensor is not independent of each other. In [11], we assumed that there are a small number of essential hidden features that characterize driving behavior, and each concrete measured sensor information can be regarded as time series data that is generated by performing a nonlinear transformation to the essential hidden features. For illustrative purpose, let us assume that there are three essential hidden features here, and they relate to “acceleration”, “changes in the directions” and “velocity of the vehicle”, respectively. For example, the observed sensor information “longitudinal acceleration” is considered to be generated by hidden features related to acceleration and changes in the directions via nonlinear transformation [12]. Note that changes in the direction cause the friction between the tire and the ground and decreases longitudinal acceleration when a driver turns the steering to go right or left. Moreover, some of the observed sensor information share the same hidden feature. For example, engine speed, speed meter and speed of wheels are generated by hidden feature associated with the velocity of the vehicle. One type of sensor information can be generated by fusing several hidden features. For example, the yaw rate is generated by fusing the hidden features associated with the velocity of the vehicle and changes in the driving direction.

The important problem for visualization is how to extract the essential features from observed driving behavior data, which contain redundant information and nonlinear relationship between different sensor information. To address this problem, we require a method that automatically extracts the essential hidden features and filters out any redundant sensor information. The method should extract the necessary and sufficient features from high-dimensional driving behavior data automatically.

In this study, we propose a driving behavior visualization method based on a feature extraction method using a deep sparse autoencoder (DSAE). We call the visualization method a *driving color map*. We reported a brief description of this method and preliminary results in [13]. In this paper, we describe a complete description and sufficient experimental evaluation of our method and demonstrate the validity of our proposed method based on several experiments. We show that the visualization obtained by our proposed method performs better than other methods. It is shown that the proposed method can represent some complex driving behaviors more clearly than other baseline methods in particular. We also present some examples of applications of the driving color map, e.g., we detected some interesting patterns in observed driving behavior by using driving color maps. The method also successfully estimated the hidden features and visualized the driving behavior on a public road using DSAE trained by using driving behavior data measured on factory circuit.

This suggests that our proposed visualization method has high generalization performance and can be used for practical use.

II. BACKGROUND

In this section, we describe a short survey about visualization methods in the field of intelligent vehicles and feature extraction methods for multivariate time series data.

A. Visualization in Intelligent Vehicles and Intelligent Transportation Systems

Visualization of driving behavior is crucially important to improve driving skill and to investigate hazardous driving behavior. However, there have been few studies of the visualization of driving behavior data. For example, to reduce road traffic accidents, Hilton et al. presented a visualization method called *SafeRoadMaps* for communicating safety information to users [14], which visualizes the crash density at different locations based on a color heat map. Treiber et al. proposed an adaptive smoothing method with a visualization method, which visualizes the spatiotemporal dynamics of traffic patterns by using colors [15]. Huang *et al.* [16] proposed a visualization method for studying urban network centralities. The goals of the studies differ from ours.

Kilicarslan and Zheng [17] proposed a method to visualize a sequence of driving scenes by using driving videos. They manually designed a method for mapping driving videos onto a temporal profile image. This method can express driving behavior indirectly based on changes in the surrounding environment. However, it is difficult for users to infer actual driving behaviors from the compressed visual image.

Takeda *et al.* [18] proposed self-coaching system using on recorded driving data an Gaussian mixture model-based driver-behavior models. They developed a web-based driving feedback system. Their goal is similar to ours. However, they focused on risky driving behaviors and did not provide a visualization method of driving behavior itself.

In our study, we use driving behavior data which represents driving behaviors directly. The driving behavior data is so high-dimensional that it is difficult to visualize intuitively. Therefore, we use an unsupervised feature extraction method to extract low-dimensional hidden features from the high-dimensional time-series data automatically without any human intervention.

B. Feature Extraction and Deep Learning

Many unsupervised feature extraction methods can extract low-dimensional hidden features, i.e., latent time series data, automatically from high-dimensional time series data. Principal components analysis (PCA) is widely used for feature extraction [19]. PCA can find the principal components that correspond to an axis having the largest variance. However, PCA basically can produce good results when the input data follow a Gaussian distribution and essential hidden features are orthogonal to each other in the vector space. Independent component analysis (ICA) [20] can effectively extract the independent hidden features from multivariate signals. The method

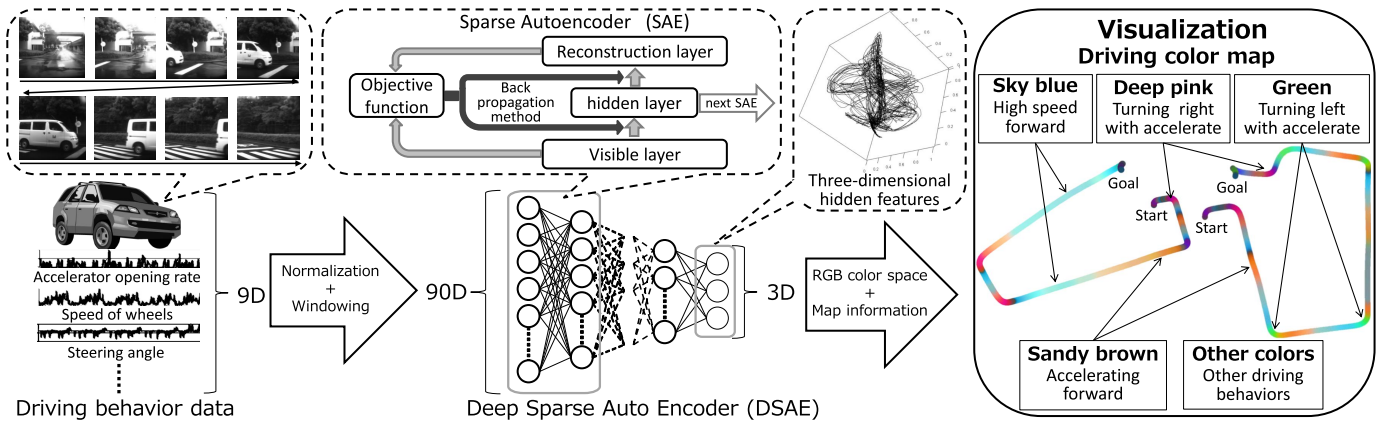


Fig. 1. The proposed method includes two processing steps: hidden feature extraction by using the deep sparse autoencoder and visualization of driving behavior in the driving color map.

assumes that the observed signals are generated via linear transformations from source signals. PCA and ICA might not be able to extract the hidden features from driving behavior data by linear transformations because vehicle dynamics and human driving behavior involve nonlinear properties.

Schölkopf *et al.* [21] proposed the kernel PCA (KPCA) method. KPCA is a feature extraction method that considers non-linear transformation. It uses a nonlinear kernel function that involves a nonlinear transformation to map the data onto a high-dimensional space. After the non-linear transformation, KPCA employs PCA to find the principal axis of the high-dimensional space. However, the computational cost of the KPCA is high when there is a large volume of driving behavior data because the kernel method must compute a Gram matrix in $\mathbb{R}^{N \times N}$, where N is the quantity of data.

In recent years, feature extraction methods that employ deep learning approaches have attracted much attention. Deep learning methods employ neural networks having a deep structure, i.e., having over three or more layers. Some deep learning methods, such as a restricted Boltzmann machine (RBM [22]), autoencoder [23], and convolutional autoencoder [24], employ pre-training before fine tuning. Pre-training comprises unsupervised learning for feature extraction. Bengio indicated that an autoencoder is similar to an RBM because both have a double-layered completely undirected graph structure, although an autoencoder can be trained more easily than an RBM [25]. In the fine tuning, the neural network having a deep structure is trained throughout all of the layers.

Few studies have used deep learning methods for driving behavior analysis. Diaz *et al.* [26] successfully used a deep learning method based on an autoencoder to model driving behaviors to estimate the energy consumption by electric vehicles. Tagawa [27] proposed structured denoising autoencoder for fault detection and applied it to driving behavior data. Our previous study showed that DSAE could be used to extract essential hidden features from driving behavior data [11]. That suggests DSAE could be used as an information filter to decrease the redundancy of driving behavior. Therefore, in this paper, we use a DSAE to extract the hidden features for visualizing driving behavior.

III. PROPOSED METHOD

In this section, we describe our proposed method in detail. Our proposed method is illustrated in Fig. 1, which comprises two steps: hidden feature extraction and visualization. In the first step, the method extracts three-dimensional features from driving behavior data. It firstly uses the normalization and windowing process to pre-process the driving behavior data, which is shown in the left part of Fig. 1. The method employ a DSAE with five encoding layers¹ to extract the three-dimensional hidden features from driving behavior data. Note that the detail of DSAE's structure using in the experiment is written in IV-A. We assume that the shallower layer can extract low level features which can represent more specific driving behaviors; the deeper layer can extract higher level features which can represent more abstract driving behavior by fusing various low level features. We expect that the hidden features of different driving behaviors are represented distinctively in the feature space. The second step is the visualization which is shown on the right part of Fig. 1. After the method calculates colors by mapping the extracted three-dimensional hidden features into the RGB color space, we place these colors in their corresponding positions on the map to generate the driving color map. The goal of the method proposed in this study is to provide an intuitive color map to make different driving behaviors represented by different colors.

A. Feature Extraction by Using DSAE

The DSAE is comprised of many sparse autoencoders (SAEs). Parameters (weight matrices and bias vectors) of each SAE is optimized layer by layer in the pre-training stage. The SAE can be viewed as a three-hierarchical directed graph, i.e., two-hierarchical undirected graph, with a visible layer, hidden layer, and reconstruction layer. It is designed to

¹We recommend to use DSAE having more than three layers for an encoder network. Based on the driving behavior data which were used in the experiment, we designed a DSAE which contains 11 layers. The dimensions of each layer are $90 \rightarrow 45 \rightarrow 22 \rightarrow 11 \rightarrow 3 \rightarrow 11 \rightarrow 22 \rightarrow 45 \rightarrow 90$. Therefore, DSAE extracts the hidden features via a five encoding layers with the dimensions of $90 \rightarrow 45 \rightarrow 22 \rightarrow 11 \rightarrow 3$. The number of layers and nodes can be changed. However, the network should be deep.

ensure that the hidden layer has the property of sparseness. Each SAE encodes input data into data on its hidden layer and then decodes the data so as to reconstruct the input data. An SAE is optimized to minimize the error between the input data and reconstructed data by using a back-propagation (BP) method. After this reconstruction error converges to a sufficiently small value, pre-training for the SAE is stopped. Each SAE outputs its hidden layer's data, i.e., hidden features, to the visible layer in the next SAE as input data. During the fine-tuning stage, the optimized parameters are initialized to train all of the layers of the DSAE. The DSAE can determine the hidden features by extracting them via the deep structure comprising tandem SAEs.

In this study, we define driving behavior data set as $\mathbf{Y} \in \mathbb{R}^{D_Y \times N_Y}$, where D_Y is the dimensionality of data; and N_Y is the amount of data, i.e., the total number of time steps of \mathbf{Y} . Observation \mathbf{y}_t at each time step t is defined as

$$\mathbf{y}_t = (y_{t,1}, y_{t,2}, \dots, y_{t,D_Y})^T \in \mathbb{R}^{D_Y}. \quad (1)$$

Before the data are given to the DSAE, normalization and windowing operation are performed. We assume that the range of observed driving behavior data in the data set \mathbf{Y} is $(-\infty, \infty)$, and we use a hyperbolic tangent function $\tanh(\cdot)^2$ as an activation function in each SAE. The range of $\tanh(\cdot)$ is $(-1, 1)$, so we should normalize the driving behavior data from $(-\infty, \infty)$ to $(-1, 1)$. We use the maximum and minimum values of each dimension to normalize the values in each dimension of the observed data into $(-1, 1)$ independently because the unit of each sensor information is different. The normalized data \mathbf{x}_t for the t -step are

$$\begin{aligned} \mathbf{x}_t &= (x_{t,1}, x_{t,2}, \dots, x_{t,D_X})^T \in \mathbb{R}^{D_Y}, \\ x_{t,d} &= 2 \left(\frac{y_{t,d} - y_{dmin}}{y_{dmax} - y_{dmin}} \right) - 1, \end{aligned} \quad (2)$$

where $y_{dmax} = \max(y_{1,d}, \dots, y_{N_Y,d})$ is the maximum value and $y_{dmin} = \min(y_{1,d}, \dots, y_{N_Y,d})$ is the minimum value of the d -th dimension of data set \mathbf{Y} . In order to extract contextual features and mitigate the ill effects of noises, we use the windowing operation. The windowed data \mathbf{v}_t have $D_V = w \times D_Y$ dimensions, which are generated by

$$\mathbf{v}_t = (\mathbf{x}_{t-w+1}^T, \mathbf{x}_{t-w+2}^T, \dots, \mathbf{x}_t^T)^T \in \mathbb{R}^{D_V} (t \geq w). \quad (3)$$

When \mathbf{v}_t moves by one step on the time axis, we can obtain a windowed data set, i.e., $\mathbf{V} \in \mathbb{R}^{D_V \times N_V}$, where $N_V = N_Y - w + 1$ is the quantity of data (time steps) in matrix \mathbf{V} .

Next, matrix \mathbf{V} is input into the DSAE as the first input data for the SAE. In addition, \mathbf{v}_t is treated as the visible layer's vector $\mathbf{v}_t^{(1)}$ for the first SAE. In the l -th SAE of the DSAE, we use the encoder function in Eq. (4) to encode a vector for the hidden layer based on the visible layer's vector $\mathbf{v}_t^{(l)}$,

$$\mathbf{h}_t^{(l)} = \tanh(\mathbf{W}_{en}^{(l)} \mathbf{v}_t^{(l)} + \mathbf{b}_{en}^{(l)}) \in \mathbb{R}^{D_H^{(l)}}, \quad (4)$$

where $D_H^{(l)}$ is the dimensionality of the hidden layer's vector in the l -th SAE, $\mathbf{W}_{en}^{(l)} \in \mathbb{R}^{D_H^{(l)} \times D_V^{(l)}}$ is a weight matrix, and

² $\tanh(\#) = (\exp^\# - \exp^{-\#}) / (\exp^\# + \exp^{-\#})$. If $\#$ is a vector, then the input of the \tanh function is each of its element.

$\mathbf{b}_{en}^{(l)} \in \mathbb{R}^{D_H^{(l)}}$ is the bias vector of the encoder. To reconstruct the data in the reconstruction layer, we use Eq. (5) to decode the data in the hidden layer:

$$\mathbf{r}_t^{(l)} = \tanh(\mathbf{W}_{de}^{(l)} \mathbf{h}_t^{(l)} + \mathbf{b}_{de}^{(l)}) \in \mathbb{R}^{D_V^{(l)}}. \quad (5)$$

In Eq. (5), $\mathbf{W}_{de}^{(l)} \in \mathbb{R}^{D_V^{(l)} \times D_H^{(l)}}$ and $\mathbf{b}_{de}^{(l)} \in \mathbb{R}^{D_V^{(l)}}$ are the weight matrix and bias vector of the decoder, respectively. To represent the visual layer's data by using the hidden layer's data, it is assumed that $\mathbf{r}_t^{(l)} = \mathbf{v}_t^{(l)}$ in the encoding-decoding processing. For the data set \mathbf{V} , the error in reconstruction between $\mathbf{r}_t^{(l)}$ and $\mathbf{v}_t^{(l)}$ is calculated as the objective function Eq. (6).

$$\begin{aligned} O(\mathbf{V}^{(l)}) &= \frac{1}{2N_V} \sum_{t=1}^{N_V} \|\mathbf{r}_t^{(l)} - \mathbf{v}_t^{(l)}\|_2^2 + \frac{\alpha}{2} (\|\mathbf{W}_{en}^{(l)}\|_2^2 + \|\mathbf{W}_{de}^{(l)}\|_2^2) \\ &\quad + \beta \sum_{i=1}^{D_H^{(l)}} \text{KL}(\omega \| \bar{h}_i^{(l)}). \end{aligned} \quad (6)$$

The average value of the squared error between all of the input data and the reconstructed data is presented by $\frac{1}{2N_V} \sum_{t=1}^{N_V} \|\mathbf{r}_t^{(l)} - \mathbf{v}_t^{(l)}\|_2^2$. To prevent over-fitting in SAE, when the elements of $\mathbf{W}_{en}^{(l)}$ and $\mathbf{W}_{de}^{(l)}$ become very large, we limit the elements of $\mathbf{W}_{en}^{(l)}$ and $\mathbf{W}_{de}^{(l)}$ with the L2 norm as a penalty term in the objective function of Eq. (6), where α can control the strength of the penalty term. We also require that the data in the hidden layer are sparse because we want to obtain more obvious features. In general, the L1 norm is used as the sparse item, but it cannot be differentiated in 0. Therefore, we use $\sum_{i=1}^{D_H^{(l)}} \text{KL}(\omega \| \bar{h}_i^{(l)})$ to calculate the sparse item in Eq. (6). The β term can control the strength of the sparse item, which is the Kullback-Leibler divergence between two Bernoulli random variables with mean ω and $\bar{h}_i^{(l)}$ [23]. If we minimize the sparse item, $\bar{h}_i^{(l)}$ will be close to ω . The sparse item is

$$\text{KL}(\omega \| \bar{h}_i^{(l)}) = \omega \log \frac{\omega}{\bar{h}_i^{(l)}} + (1 - \omega) \log \frac{1 - \omega}{1 - \bar{h}_i^{(l)}}, \quad (7)$$

where ω is the sparsity target of the hidden layer, which needs to be specified. $\bar{h}_i^{(l)}$ is the average value of the i -th dimension of the average vector $\bar{\mathbf{h}}^{(l)} \in \mathbb{R}^{D_H^{(l)}}$, which comprises $\bar{h}_i^{(l)}$:

$$\bar{h}_i^{(l)} = \frac{1}{2} \left(1 + \frac{1}{N_V} \sum_{t=1}^{N_V} h_{t,i}^{(l)} \right), \quad (8)$$

where $h_{t,i}^{(l)}$ is the i -th element of $\mathbf{h}_t^{(l)}$. In Eq. (7), the log function is inside; therefore, the range of $\omega / \bar{h}_i^{(l)}$ must be in $(0, +\infty)$. However, the range of the activation function $\tanh(\cdot)$ is $(-1, 1)$, so we cannot calculate $\log(\omega / \bar{h}_i^{(l)})$. To address this problem, we scale $\bar{h}_i^{(l)}$ from $(-1, 1)$ to $(0, 1)$. There is another advantage of using this sparse term in our method because when the sparsity set is at 0 in the \tanh space, this can make the average value of each data's dimension in the hidden layer remain at the center of the feature space. Thus, in our visualization method—driving color map, the generated colors do not tend to appear biased, e.g., reddish, bluish, or

TABLE I
THE OBSERVED SENSOR INFORMATION

Accelerator opening rate [%]	Brake master-cylinder pressure [MPa]	Steering angle [deg]	Speed of wheels [km/h]	Speed meter [km/h]
Engine speed [rpm]	Longitudinal acceleration [m/s ²]	Transverse acceleration [m/s ²]	Yaw rate [rad/s]	

others. We introduce the specific visualization methods in the next subsection.

Finally, we use the BP method [28] to minimize the objective function and train the SAE. The BP method requires the partial differentiation of the weight matrices $\mathbf{W}_{en}^{(l)}$, $\mathbf{W}_{de}^{(l)}$ and biases $\mathbf{b}_{en}^{(l)}$, $\mathbf{b}_{de}^{(l)}$ for the objective function Eq. (6). The partial differential equations for the decoder are

$$\frac{\partial O^{(l)}}{\partial \mathbf{W}_{de}^{(l)}} = \frac{1}{N_v} \sum_{t=1}^{N_v} \mathbf{h}_t^{(l)} \boldsymbol{\gamma}_t^T + \alpha \mathbf{W}_{de}^{(l)}, \quad (9)$$

$$\frac{\partial O^{(l)}}{\partial \mathbf{b}_{de}^{(l)}} = \frac{1}{N_v} \sum_{t=1}^{N_v} \boldsymbol{\gamma}_t, \quad (10)$$

where the vector $\boldsymbol{\gamma}_t \in \mathbb{R}^{D_v^{(l)}}$ is

$$\boldsymbol{\gamma}_t = \text{diag}((r_{t,1}^{(l)})^2 - 1, (r_{t,2}^{(l)})^2 - 1, \dots, (r_{t,D_v^{(l)}}^{(l)})^2 - 1)(\mathbf{v}_t^{(l)} - \mathbf{r}_t^{(l)}).$$

The partial differential equations for the encoder are

$$\frac{\partial O^{(l)}}{\partial \mathbf{W}_{en}^{(l)}} = \frac{1}{N_v} \sum_{t=1}^{N_v} \mathbf{v}_t^{(l)} \boldsymbol{\xi}_t^T + \alpha \mathbf{W}_{en}^{(l)}, \quad (11)$$

$$\frac{\partial O^{(l)}}{\partial \mathbf{b}_{en}^{(l)}} = \frac{1}{N_v} \sum_{t=1}^{N_v} \boldsymbol{\xi}_t, \quad (12)$$

where the vector $\boldsymbol{\xi}_t \in \mathbb{R}^{D_H^{(l)}}$ is

$$\boldsymbol{\xi}_t = \text{diag}(1 - (h_{t,1}^{(l)})^2, 1 - (h_{t,2}^{(l)})^2, \dots, 1 - (h_{t,D_H^{(l)}}^{(l)})^2) \\ \times (\mathbf{W}_{de}^{(l)} \boldsymbol{\gamma}_t) + \beta \boldsymbol{\epsilon},$$

and the i -th element of the vector $\boldsymbol{\epsilon} \in \mathbb{R}^{D_H^{(l)}}$ is

$$\epsilon_i = \frac{1 - \omega}{1 - \bar{h}_i^{(l)}} - \frac{\omega}{\bar{h}_i^{(l)}}.$$

The equations for the biases $\mathbf{b}_{en}^{(l)}$, $\mathbf{b}_{de}^{(l)}$ and weight matrices $\mathbf{W}_{en}^{(l)}$, $\mathbf{W}_{de}^{(l)}$ are updated using Eqs. (13), (14), (15), and (16):

$$\mathbf{W}_{en}^{+(l)} \leftarrow \mathbf{W}_{en}^{(l)} - \lambda_{en} \frac{\partial O(\mathbf{V}^{(l)})}{\partial \mathbf{W}_{en}^{(l)}}, \quad (13)$$

$$\mathbf{b}_{en}^{+(l)} \leftarrow \mathbf{b}_{en}^{(l)} - \lambda_{en} \frac{\partial O(\mathbf{V}^{(l)})}{\partial \mathbf{b}_{en}^{(l)}}, \quad (14)$$

$$\mathbf{W}_{de}^{+(l)} \leftarrow \mathbf{W}_{de}^{(l)} - \lambda_{de} \frac{\partial O(\mathbf{V}^{(l)})}{\partial \mathbf{W}_{de}^{(l)}}, \quad (15)$$

$$\mathbf{b}_{de}^{+(l)} \leftarrow \mathbf{b}_{de}^{(l)} - \lambda_{de} \frac{\partial O(\mathbf{V}^{(l)})}{\partial \mathbf{b}_{de}^{(l)}}, \quad (16)$$

where λ_{en} and λ_{de} are the learning rates of encoder and decoder, which can control the intensity of each update. Setting the learning rate is difficult, but an appropriate learning rate

can make the SAE converge faster. In this paper, we use a line search to find a optimal value of the learning rate after calculating the differential when $O(\mathbf{V}^{(l)})$ is the minimum in the direction of the differential. This method can automatically search the appropriate learning rate for each update.

During line search, we define the searching distance θ . The learning rate λ_* ($* \in \{en, de\}$) and searching distance θ are initialized as small positive numbers. Thus, the appropriate learning rate becomes $\lambda_*^+ = \lambda_* + \theta$. θ is updated by line search:

$$\theta^+ = \begin{cases} -0.5\theta & (O^+(\mathbf{V}^{(l)}) > O(\mathbf{V}^{(l)})) \\ \theta & (O^+(\mathbf{V}^{(l)}) \leq O(\mathbf{V}^{(l)})) \end{cases}, \quad (17)$$

where $(\cdot)^+$ is the value that has been updated. We set the stop condition of line search as the point when the change in the error $|O^+(\mathbf{V}^{(l)}) - O(\mathbf{V}^{(l)})|$ is less than a certain threshold. We also use the same criterion to stop the updating of the weight matrices and biases in the error equation (6). Therefore, we can obtain an optimized l -th SAE. To create a DSAE, we use the feature matrix $\mathbf{H}^{(l)} = (\mathbf{h}_1^{(l)}, \dots, \mathbf{h}_{N_v}^{(l)}) \in \mathbb{R}^{D_H^{(l)} \times N_v}$ of the l -th SAE as the visible layer $\mathbf{V}^{(l+1)} \in \mathbb{R}^{D_H^{(l)} \times N_v}$ for the next $(l+1)$ -th SAE ($\mathbf{V}^{(l+1)} = \mathbf{H}^{(l)}$). In summary, the t -th time step data \mathbf{v}_t is mapped to extract the features $\mathbf{h}_t^{(final)}$ by stacking many SAEs, e.g., the 1st SAE: $\mathbf{v}_t \mapsto \mathbf{h}_t^{(1)} \mapsto \mathbf{r}_t^{(1)}$, the 2nd SAE: $\mathbf{h}_t^{(1)} \mapsto \mathbf{h}_t^{(2)} \mapsto \mathbf{r}_t^{(2)}$, \dots , the final SAE: $\mathbf{h}_t^{(final-1)} \mapsto \mathbf{h}_t^{(final)} \mapsto \mathbf{r}_t^{(final)}$.

After pre-training phase, we stack many SAE into a DSAE, which is a deep neural network with an encoder-decoder structure. This fine-tuning stage maps the t -th time step data \mathbf{v}_t to reconstruct itself by encoding process: $\mathbf{v}_t \mapsto \mathbf{h}_t^{(1)} \mapsto \mathbf{h}_t^{(2)} \mapsto \dots \mapsto \mathbf{h}_t^{(final-1)} \mapsto \mathbf{h}_t^{(final)}$, and decoding process: $\mathbf{h}_t^{(final)} \mapsto \mathbf{r}_t^{(final)} \mapsto \dots \mapsto \mathbf{r}_t^{(2)} \mapsto \mathbf{r}_t^{(1)}$. The $\mathbf{h}_t^{(final)}$ is the extracted hidden feature of \mathbf{v}_t , which is used for visualization. We also utilize the BP method to train the DSAE during fine tuning.

B. Driving Behavior Visualization by the Driving Color Map

In the second step, we visualize the driving behavior by using the extracted hidden three-dimensional features. The visualization method is called the driving color map, which comprises a colored trajectory shown on a road map that represents the extracted features. In this section, we assumed that nine types of sensor information are obtained via a CAN as shown in Table I although the proposed method can be applied other to set of sensor data. It was suggested that the three-dimensional hidden features are almost sufficient to represent driving behaviors in [11]. In addition, the RGB color space is a three-dimensional space. Based on the two facts, we extract the three-dimensional hidden features to visualize driving behavior in this study. We obtain the colors in

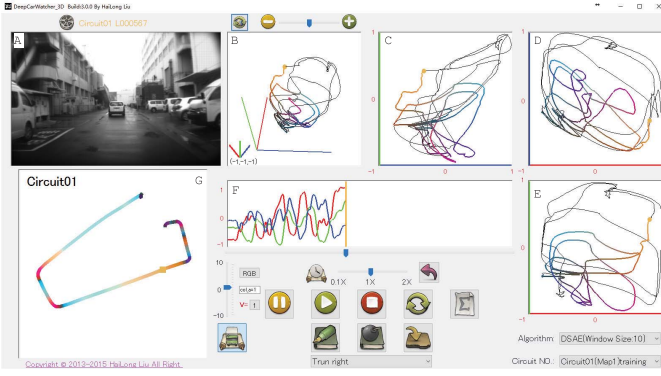


Fig. 2. Deep Car Watcher by which users can review the driving videos (A), and the extracted three-dimensional hidden features in the feature space (B, C, D, and E) and time space (F), which can also review the driving color map (G) simultaneously.

the RGB color space corresponding to the three-dimensional hidden features by a simple scaling method. When the range of the RGB color space is $[0, 1]^3$, we normalize the three-dimensional hidden features into $[0, 1]^3$. In summary, we map the three-dimensional hidden features to the RGB space by

$$rgb_{t,d} = \frac{h_{t,d}^{(final)} - h_{min_d}^{(final)}}{h_{max_d}^{(final)} - h_{min_d}^{(final)}}, \quad (18)$$

where $rgb_{t,d}$ is a d -th element of a three-dimensional vector in the RGB space that represents the driving behavior at the t -th time step. $h_{t,d}^{(final)}$ is the d -th element of the extracted three-dimensional hidden feature's vector at the t -th time step. The terms $h_{min_d}^{(final)}$ and $h_{max_d}^{(final)}$ are the minimum and maximum values of the d -th dimension in $\mathbf{H}^{(final)}$, respectively. Finally, we obtain a driving color map by placing the colors in the corresponding positions on the map.

To monitor the visualization results, we develop a program called *Deep Car Watcher*³ (Fig. 2), which can simultaneously review the driving videos, extracted hidden features, driving color map, and time-series. A point indicates the current vehicle location on the driving color map.

IV. VISUALIZATION EXPERIMENTS

A. Experimental Conditions

In this experiment, we verified that hidden features extracted using DSAE were better than those employed by other methods for visualizing driving behavior. We employed PCA, FastICA [29], KPCA with the RBF kernel and SAE as comparative methods to extract three-dimensional features from driving behavior data and generate driving color maps.

To obtain the driving behavior data, we asked a participant to drive an experimental vehicle through two courses at a company's factory. The participant drove each course five times. During the experiment, the factory was in normal operation. Therefore, the car encountered different situations on the courses, such as pedestrians, other moving vehicles, and parked vehicles. The Circuits 1–5 correspond to the first

course and Circuits 6–10 correspond to the second course. We observed 12958 frames of driving behavior data in total at a frame rate of 10 fps. Each data frame could include the nine sensor information captured via the CAN (see Table I).

As a working hypothesis, we assume that there are three essential hidden features, and they roughly correspond to “vehicle acceleration”, “vehicle velocity” and “changes in the driving direction”. We considered that the accelerator opening rate, brake master-cylinder pressure and longitudinal acceleration were mainly related to the acceleration of the vehicle; the speed of wheels, speed meter, and engine speed were mostly related to the velocity of the vehicle; the steering angle could represent a change in the driving direction; and the longitudinal acceleration and yaw rate included information about acceleration of the vehicle and changes in the driving direction. Thus, we prepared a simple three-dimensional feature called VV'S, i.e., the speed of wheels, the differential of the speed of wheels on time, and steering angle, which are expected to relate to the three essential hidden features assumed above. Then we generated driving color maps by using VV'S in addition to the above-mentioned comparative methods.

We used the VV'S and the three-dimensional hidden features extracted using PCA, FastICA, KPCA, SAE and DSAE to generate the driving color maps for comparison. These driving color maps were compared with maps obtained using the hidden features extracted by the DSAE. Except for the VV'S, each feature extraction method employed the windowing process with a time window size $w = 10$, i.e., one second. Then the nine-dimensional driving behavior data \mathbf{y}_t becomes to the ninety-dimensional windowed data \mathbf{v}_t . We employed PCA, FastICA, KPCA, and SAE to extract three-dimensional hidden features from the windowed data. When DSAE was used to extract three-dimensional hidden features from the windowed data, the dimensions of each encoding layer were set to be 90, 45, 22, 11, and 3. The number of nodes of each layer was set to be a half of that of its previous layer approximately. We set the hyper-parameter ϖ of the RBF kernel $k(x, x') = \exp(-\varpi \|x - x'\|^2)$ to 0.1 for KPCA by our experience of many experiments before. In addition, the parameters α and β of the SAEs and DSAEs were set empirically to $\alpha = 0.03$, $\beta = 0.7$, but without using a specific parameter tuning method. To generate a driving color map with various colors, we considered that the average value of the hidden features should be located in the center of the RGB color space. Therefore, ω was set to 0.5 because the center of each axis of the RGB color space is 0.5.

B. Visualization Results

In this section, the direct observation of the visualization results is reported, first before we conduct quantitative evaluation in the following sections. The driving color maps generated by each method for Circuits 1 and 6 are shown as examples in Fig. 3. This figure shows that the colors in the driving color maps produced using the VV'S contained a significant amount of noise, and the colors were not as rich as those generated using the other methods. Thus, the driving

³For illustrative purpose, a video of Deep Car Watcher is uploaded and available at : <https://www.youtube.com/watch?v=54JWzoFn13E>

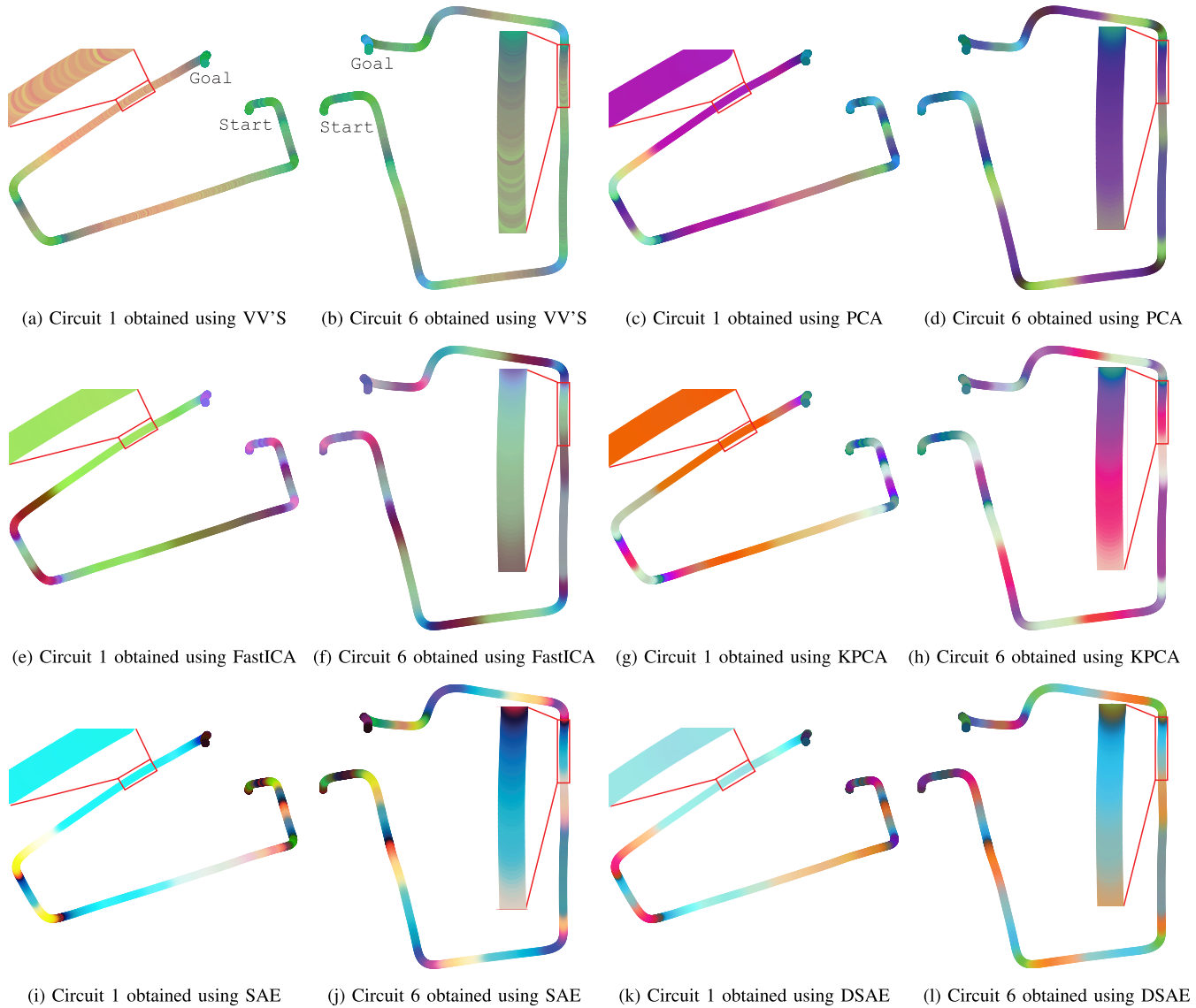


Fig. 3. The driving color maps for Circuits 1 and 6 are shown as examples, which were generated based on the VV'S and the three-dimensional hidden features extracted using PCA, FastICA, KPCA, SAE, and DSAE. (a) Circuit 1 obtained using VV'S. (b) Circuit 6 obtained using VV'S. (c) Circuit 1 obtained using PCA. (d) Circuit 6 obtained using PCA. (e) Circuit 1 obtained using FastICA. (f) Circuit 6 obtained using FastICA. (g) Circuit 1 obtained using KPCA. (h) Circuit 6 obtained using KPCA. (i) Circuit 1 obtained using SAE. (j) Circuit 6 obtained using SAE. (k) Circuit 1 obtained using DSAE. (l) Circuit 6 obtained using DSAE.

color maps generated using the extracted hidden features with windowing processing could smooth the noise and represent driving behaviors better than the observed data (VV'S).

We detected several simple and complex driving behaviors in the driving videos. The most obviously found driving behaviors were typical and simple driving behaviors, such as high speed forward, stopping the vehicle, accelerating forward, right rear reversing, and left rear reversing. We also found complex driving behaviors which involved several simple driving behaviors occurred simultaneously. For example, simple driving behaviors such as turning right and accelerating could be combined to yield complex driving behavior of “turning right while accelerating”. Similarly, turning left and accelerating could be combined to yield “turning left while accelerating”.

We observed and compared the driving color maps generated by each method to find their representative colors

corresponding to each driving behavior. The representative color for each driving behavior is shown in Table II. Note that the representative colors were picked up manually and subjectively, and Table II is shown for illustrative purpose. The objective and quantitative evaluation is performed in the following sections.

It was observed that the proposed visualization method using DSAE generated distinctly different colors for different driving behaviors: high speed forward (light blue), stopping vehicle (dark blue), accelerate forward (orange), right rear reversing (dark purple), left rear reversing (green), turning right with constant speed (purple), turning left with constant speed (yellow-green), turning right while accelerating (pink), and turning left while accelerating (light green).

In contrast, it was found that other methods generated similar colors for different driving behaviors. The similar colors for different driving behaviors are underlined in Table II. The similar colors obtained by each method are indicated by the

TABLE II
REPRESENTATIVE COLORS OF SIMPLE AND COMPLEX DRIVING BEHAVIORS OBTAINED USING THE VV'S AND THE FEATURES EXTRACTED BY PCA, FASTICA, KPCA, SAE, AND DSAE

Methods	Simple driving behaviors					Complex driving behaviors			
	High speed forward	Stopping vehicle	Accelerating forward	Right rear reversing	Left rear reversing	Turning right with constant speed	Turning left with constant speed	Turning right while accelerating	Turning left while accelerating
VV'S									
PCA									
FastICA									
KPCA									
SAE									
DSAE									

TABLE III
F-MEASURE OF SEPARABILITY IN COLOR SPACE OBTAINED BY USING LINEAR SVM FOR SIMPLE AND COMPLEX DRIVING BEHAVIORS

Methods	Simple driving behaviors								Complex driving behaviors				
	High speed forward	Stopping vehicle	Accelerating forward	Turning right	Turning left	Right rear reversing	Left rear reversing	Ave.	Turning right with constant speed	Turning left with constant speed	Turning right while accelerating	Turning left while accelerating	Ave.
VV'S	0.753	0.163	<u>0.442</u>	0.377	0.234	0.886	0.818	0.525	0.108	0.221	0.230	0.106	0.140
PCA	0.587	0.382	0.427	0.769	0.781	0.768	0.407	0.589	0.126	0.220	0.795	0.825	<u>0.582</u>
FastICA	0.842	0.296	0.410	<u>0.740</u>	<u>0.750</u>	<u>0.878</u>	0.887	<u>0.686</u>	<u>0.144</u>	0.199	<u>0.799</u>	0.739	<u>0.565</u>
KPCA	<u>0.878</u>	0.369	0.468	0.400	0.158	0.092	0.039	0.343	0.088	0.062	0.188	0.087	0.119
SAE	0.882	<u>0.514</u>	<u>0.351</u>	0.659	0.601	0.729	0.609	0.621	0.119	<u>0.298</u>	0.782	<u>0.865</u>	<u>0.582</u>
DSAE	0.823	0.552	0.433	<u>0.747</u>	0.611	0.820	<u>0.849</u>	0.691	0.148	0.362	0.858	0.883	0.626

same type of underline (or). For example, several driving behaviors corresponded to similar colors by using the VV'S, where the driving behaviors related to turning right were represented in green, such as right rear reversing () , turning right with constant speed () and turning right while accelerating () . Driving behaviors related to turning left were shown by sky blue, e.g., turning left with constant speed () and turning left while accelerating () . This showed that the identification of driving behaviors was not highly successful by using the VV'S and it was difficult to visualize complex driving behaviors. Using PCA, turning right with constant speed () and right rear reversing () were both represented by blue, while accelerating forward () and turning left while accelerating () were shown in green. Using FastICA, light purple represented stopping vehicle () and right rear reversing () . Driving behaviors such as stopping vehicle () , turning right with constant speed () and turning left with constant speed () were shown by a similar color by using KPCA. As well as the colors generated by KPCA of accelerating forward () and turning right while accelerating () were also alike. When using SAE, the corresponding colors for right rear reversing () and left rear reversing () were difficult to distinguish. By contrast, using DSAE, different colors corresponded to distinct driving behaviors.

These results qualitatively suggest that the driving color maps obtained using DSAE made it easier for users to distinguish each driving behavior using the driving color maps than other methods.

C. Numerical Evaluation of Visual Separability in Driving Color Maps

We assessed whether different driving behaviors can be distinguished in each color space. We assumed that if the corresponding color of each driving behavior was linearly separable in the color space, then it could also be readily distinguished by a human. Thus, we used a binary support vector machine (SVM) with a linear kernel⁴ to evaluate the separability of the visualized driving behaviors.

We used the F-measure to evaluate the generalization performance of the generated colors based on leave-one-out cross-validation. In particular, we selected the driving behavior data from one circuit as the test set and the driving behavior data from the remaining nine circuits as the training set. We used the training set to train the feature extraction model and for extracting three-dimensional hidden features. Next, we used the hidden features extracted from the training set to train an SVM with a linear kernel for recognizing a specific driving behavior listed in Table III. We then inputted the test set into the trained feature extraction model to estimate the hidden features. The estimated hidden features were inputted into the trained SVM to estimate whether the estimated hidden features belonged to the specified driving behavior. We selected the driving behavior data from another circuit as the test set and repeated the training and estimation processes described above. Thus, we performed ten trials for each driving behavior

⁴We used *kernelab* on R 3.2.3 32bit as a library for SVM implementation.

shown in Table III for one feature extraction method. The segments and truth labels corresponding to each driving behavior shown in Table III was prepared in advance.

We counted the true positive (TP), false positive (FP), and false negative (FN) rates in the ten trials, and we calculated the F-measure as follows:

$$F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}, \quad (19)$$

where $\text{Precision} = \text{TP}/(\text{TP}+\text{FP})$ and $\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$.

We selected several simple driving behaviors and complex driving behaviors, as shown in Table III. The F-measures obtained by the linear SVM for simple and complex driving behaviors by using DSAE and comparative methods are listed in Table III. We mark the highest F-measure by underline and bold font for each driving behavior. Meanwhile, the second place of F-measure is marked by underline only.

First, we focus on simple driving behaviors. Even though the VV'S obtained good performance for some of the simple driving behaviors, it obtained worse results than other methods for complex driving behaviors. Thus, the VV'S can represent simple driving behaviors to some extent, but they are not suitable for complex driving behaviors. DSAE had the highest F-measure for stopping vehicle, and the second highest F-measures for turning right, and left rear reversing. The highest average F-measure was obtained by DSAE. PCA had the highest F-measures for turning right and turning left. FastICA could represent the right rear reversing and left rear reversing driving behaviors well, and it also obtained the second highest average F-measure. The results obtained using KPCA were better for accelerating forward, but it obtained the worst performance in average. When we used SAE, the F-measure was highest for high speed forward, and the F-measures for the other driving behaviors were intermediate.

Next, we focused on the complex driving behaviors. Our results clearly demonstrated that DSAE obtained better F-measures for all of the complex driving behaviors than others. Thus, it was shown that the driving color maps using DSAE could represent complex driving behaviors more clearly than the other methods.

The results of this experiment and our qualitative observation shown in the previous subsection were consistent. In the next section, we performed a subjective evaluation experiment to determine whether the driving color map based on the DSAE helps users to review and understand driving behavior intuitively or not.

D. Subjective Evaluation of the Driving Color Map

To clarify whether our proposed method allowed users to identify different driving behaviors easily or not, we conducted a subjective evaluation experiment. We asked nineteen participants to complete a questionnaire in order to evaluate the driving color maps obtained using DSAE and other methods. All, except two of these participants, had a driver's license. To perform subjective evaluations, we developed a modified version of Deep Car Watcher, as shown in Fig. 4, where A~E indicate the names of our proposed visualization method and four other methods used for comparative purposes.



Fig. 4. The modified version of Deep Car Watcher for subjective evaluation, which shows driving videos and the driving color maps obtained using PCA (A), FastICA (B), KPCA (C), SAE (D), and DSAE (E). This software does not show the name of the methods, but instead it indicates them as A~E to label the driving color maps.

In this subjective evaluation, we provided participants with the modified version of Deep Car Watcher (Fig. 4).

We considered that each participant had a different way of distinction of driving behaviors. For example, some people may consider that accelerating from a standstill and accelerating from a constant speed cruise are different driving behaviors, whereas others consider them as the same. The experiment consisted of two parts. First, we asked the participants to identify driving behavior patterns that they could recognized by observing driving videos. The driving behaviors recognized by the participants were used in the next part of the evaluation task. Second, we focused on the quality of the visualization methods, and determined if each method help participants to distinguish the driving behavior patterns. We asked the participants to compare driving behavior patterns and colors on the driving color maps and to compare the five driving color maps. We asked them to rank the driving color maps according to the criterion of “the same color represents the same driving behavior”. It took about one hour for each participant to complete the questionnaire. Half of the time was spent searching for driving behaviors in the driving videos. All of the participants identified the simple driving behaviors, but only eight participants could identify the simple and complex driving behaviors considered in the previous experiments (see Table III). A Few participants found specific driving behaviors such as “coasting deceleration after turning right.” This suggested that different people have different understanding about distinctive driving behavior.

The results of the ranking are shown as histograms in Fig. 5. The vertical axis represents the order, and the horizontal axis represents the number of participants who selected a specific order. Figure 5 shows that the counts for the driving color map produced using DSAE ranked first compared with the others. The thick solid line in the histogram indicates the median value of the rankings. The driving color maps obtained using the SAE and DSAE had the highest median values.

We compared the ranking results by using statistical tests. Given that the ranking results followed an ordinal scale, we employed the Wilcoxon rank-sum test [30], which is a non-parametric statistical test. We set the null hypothesis H_0 of Wilcoxon rank-sum test as participants gave the same ranking to the driving color maps produced using two different

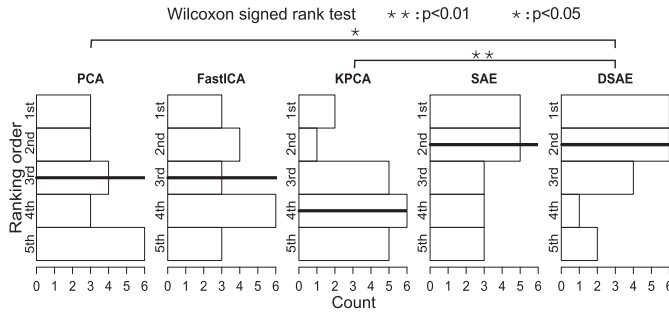


Fig. 5. Rankings for the driving color maps obtained using PCA, FastICA, KPCA, SAE, and DSAE given by nineteen subjects. The thick solid line shows the median value of the rankings.

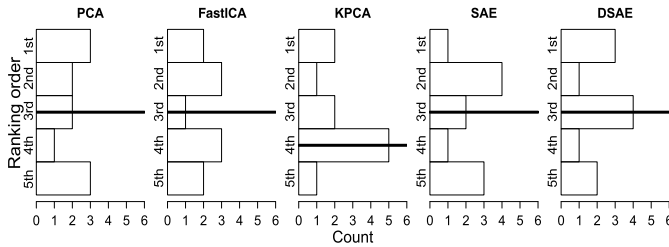


Fig. 6. Rankings for the driving color maps obtained using PCA, FastICA, KPCA, SAE, and DSAE given by eleven subjects who only identified the simple driving behaviors in driving videos. There were no significant differences among all the methods according to the Wilcoxon rank-sum test. The thick solid line shows the median value of the rankings.

methods. The confidence level was set to 95% in this test. Figure 5 shows that the p-value was less than 0.05 according to the ranking results for the driving color maps obtained using DSAE and PCA while the p-value was less than 0.001 for the comparison between DSAE and KPCA. Thus, we rejected H_0 for the driving color maps according to the comparisons of DSAE and PCA, and DSAE and KPCA, which showed that participants gave significantly different rankings. However, we were unable to reject H_0 based on the comparisons of the other pairs of methods.

We found that eight participants recognized some complex driving behaviors, which showed that more than half of the participants evaluated the visual results based only on the simple driving behaviors, so these participants gave negative evaluations for the visualized complex driving behaviors. To address this issue, we divided the participant ranking results into two groups: the ranking results given by the eleven participants who only recognized simple driving behaviors and those given by the eight participants who recognized both simple and complex driving behaviors. Figure 6 shows the experimental results for the eleven participants who only recognized simple driving behaviors. There were no significant differences among the results according to the Wilcoxon rank-sum test. Next, Fig. 7 shows the experimental results for the eight participants who recognized both the simple and complex driving behaviors. They demonstrate that the driving color maps obtained using SAE had the highest median ranking value, followed by DSAE. It should be noted that the driving color maps obtained using DSAE were always ranked in the first and second places according to the results. We also

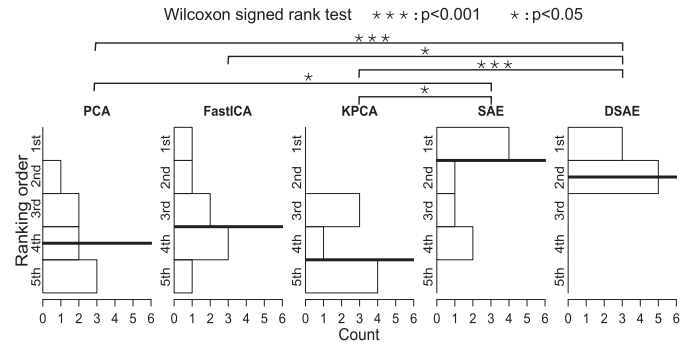


Fig. 7. Rankings for the driving color maps obtained using PCA, FastICA, KPCA, SAE, and DSAE given by eight subjects who identified the simple and complex driving behaviors in driving videos. The thick solid line shows the median value of the rankings.

used the Wilcoxon rank-sum test to verify these results. The p-values for the comparisons of the rankings for the driving color maps obtained using DSAE and PCA, and DSAE and KPCA were less than 0.001. In addition, the p-values for the comparisons of the rankings for the driving color maps obtained using DSAE and ICA, SAE and PCA, and SAE and KPCA were less than 0.05. Thus, pairs of driving color maps received significantly different rankings from the participants. We can reject H_0 for the comparisons given above, but not for DSAE and SAE. In summary, the driving color maps obtained using SAE and DSAE received higher rankings from the eight participants who recognized both simple and complex driving behaviors. It was clearly shown that the driving color maps using DSAE had a high capacity for the visual representation of both simple and complex driving behaviors.

Thus, we verified that our proposed method, driving color maps using DSAE, performed well at representing different driving behaviors with different colors. It helped participants to distinguish driving behaviors from visualization results.

V. APPLICATIONS OF DRIVING COLOR MAPS USING DSAE

After verifying the effectiveness of visualization of driving color maps using DSAE, we focused on the practical applications of the proposed method.

A. Detection of Interesting Patterns in Driving Behaviors

To determine interesting patterns in driving behaviors, we compared the driving color maps obtained using DSAE based on multiple circuits of the same course. One of the interesting patterns was “pedestrian on a pedestrian crossing in front of the experimental vehicle,” as shown in Fig. 8. In this figure, the red circles denote the pattern with pedestrians on a pedestrian crossing or with a car in front. In contrast, the blue boxes denote the pattern without pedestrians on the crossing. When a pedestrian was on the pedestrian crossing, a dark gray-blue color appeared suddenly on the driving color map. By contrast, the color changed in a smooth manner without a pedestrian. In the driving video of circuit 1, another car was in front of our experimental vehicle and people were on the pedestrian

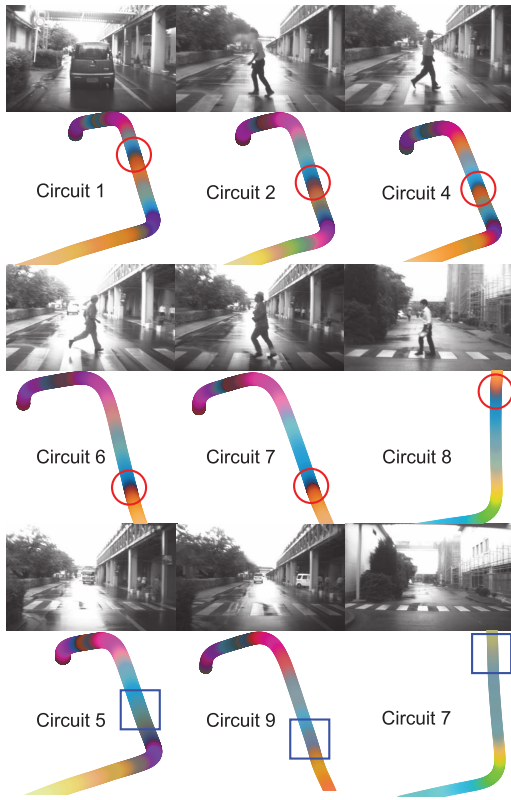


Fig. 8. Samples of visualization results: the top two parts of the figure show that a pedestrian was on a pedestrian crossing or a car was in front, and the lower part of the figure shows situations where there are no pedestrians on the crossing. These results show the visualization results were clearly affected by the difference of driving behaviors caused by different situations.

crossing in front of the car. This suggests that the driving color map helps users to notice actual changes in driving behaviors on the circuit. This will enable users to identify interesting and precarious driving behaviors easily, particularly when they monitor driving behaviors and investigate traffic accidents.

B. Visualization of Driving Behaviors on a Public Road

The driving behavior data obtained on the course inside the company's factory were used in the experiments described above. To verify if our proposed visualization method is suitable under driving conditions on a public road, and to verify its generalization performance, we asked another participant to drive the experimental vehicle around Hidaka Park in Nagoya, Japan. We obtained driving behavior data for 2064 steps, which included the nine types of sensor information that we used previously. We input the new data into our method, which was trained using the data described in previous subsections. The driving color map obtained for public road around Hidaka Park is shown in Fig. 9.

We used Deep Car Watcher to identify driving behaviors in the driving videos and to assign their corresponding colors. We compared the corresponding colors generated from the data for the factory road and the data for the public road, and these colors are shown in Table IV. There was slight difference for turning left with constant speed between the colors in the two data sets, but the rest were similar. This difference occurred because most of the corners on the factory road were right



Fig. 9. Visualization result of the driving behavior on the public road around Hidaka Park by using the trained DSAE.

TABLE IV
REPRESENTATIVE COLORS OF DRIVING BEHAVIORS OBTAINED FOR THE FACTORY ROAD AND THE PUBLIC ROAD

Driving environments	driving behaviors				
	High speed forward	Stopping vehicle	Accelerating forward	Turning left with constant speed	Left rear reversing
Factory road					
public road					

angles, whereas some of the corners on the public road did not require a great turn of the steering wheel. We found that the maximum steering angle was $620 [deg]$ and the minimum steering angle was $-624 [deg]$ for the factory data after examining the raw data. For the public road data, the maximum steering angle was $486 [deg]$ and the minimum steering angle was $-297 [deg]$. This explains why the colors generated from the factory data and public road data were slightly different. In general, it was suggested that our proposed visualization method is effective for a public road, and its performance was highly robust and applicable to data obtained in a new environment.

VI. CONCLUSION

In this study, we proposed a visualization method for driving behaviors called the driving color map based on the hidden features extracted using DSAE which is type of unsupervised deep learning methods. We employed DSAE to extract the three-dimensional hidden features from driving behavior data. The method obtained the corresponding colors by projecting the extracted three-dimensional hidden features to the RGB color space. Diving color map was obtained by placing these colors in the corresponding positions on the map.

Based on numerical evaluations and subjective experiments, we demonstrated that the proposed visualization method, driving color map using DSAE, allows people to distinguish different driving behaviors more easily than the other

comparative methods. Finally, we presented some examples of the practical application of the proposed method and verified its suitability for use on public roads.

Future challenges are as follows. The DSAE is an unsupervised feature extraction model. Therefore, there are rotational degrees of freedom between the feature spaces. This suggests that the visualization result can yield different visualization results on rotation in the color space even if it uses the same data. To clarify how to deal with the arbitrariness, i.e., the degree of freedom, appropriately is one of our future challenges. To develop practical support systems using the proposed visualization method is also our future challenge. In this paper, we did not evaluate our method from the viewpoint of a self-coaching system though one of the purposes of the visualization is to help users to review their driving behavior and improve them. Takeda *et al.* [18] developed a self-coaching system using driving behavior data and a GMM-based driver behavior models. They showed their system's potential benefit through an experiment. Developing a self-coaching system using our proposed method and showing its benefit are also our future challenge.

Exploring the possible application of DSAE-based feature extraction for intelligent vehicle is another future challenge. We suggest that our proposed method can be used to provide information about previous driving behavior. For example, a teacher at a driving school could share a driving color map with a student after they have completed a driving lesson. If the teacher finds some colors that correspond to dangerous driving behaviors, they could give the student suitable guidance while showing it. As Another example, when an accident investigator need to search for clues in a large amount of raw data, he could use the driving color map to quickly and easily understand the driving behaviors to determine the cause of the accident. Our method does not tell whether driving behaviors are safe or dangerous, but it can still help users to look through and investigate driving behaviors.

In this paper, it is shown that DSAE has an excellent capability to extract hidden features from driving behavior data. Making use of the DSAE-based feature extraction method for various purposes is also our future challenge. We expect that the extracted hidden features can be used in various ways effectively, such as driving behavior segmentation, inference of driving intention, and to learn driving strategies from drivers.

REFERENCES

- [1] Q. Zhou and F. Wang, "Driver assisted fuzzy control of yaw dynamics for 4WD vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 425–430.
- [2] H. Berndt, S. Wender, and K. Dietmayer, "Driver braking behavior during intersection approaches and implications for warning strategies for driver assistant systems," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2007, pp. 245–251.
- [3] J. C. McCall and M. M. Trivedi, "Driver behavior and situation aware brake assistance for intelligent vehicles," *Proc. IEEE*, vol. 95, no. 2, pp. 374–387, Feb. 2007.
- [4] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1292–1304, Apr. 2011.
- [5] M. Heesen *et al.*, "Interaction design of automatic steering for collision avoidance: Challenges and potentials of driver decoupling," *IET Intell. Transp. Syst.*, vol. 9, no. 1, pp. 95–104, Jan. 2015.
- [6] C. Creusot and A. Munawar, "Real-time small obstacle detection on highways using compressive RBM road reconstruction," in *Proc. IEEE Intell. Vehicles Symp.*, Jul. 2015, pp. 162–167.
- [7] W. Khaisongkram, P. Raksincharoensak, M. Shimosaka, T. Mori, T. Sato, and M. Nagai, "Automobile driving behavior recognition using boosting sequential labeling method for adaptive driver assistance systems," in *KI 2008: Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2008, pp. 103–110.
- [8] S. Nagasaka, T. Taniguchi, K. Hitomi, K. Takenaka, and T. Bando, "Prediction of next contextual changing point of driving behavior using unsupervised Bayesian double articulation analyzer," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 924–931.
- [9] D. Aiteanu, J. Zeisler, and M. Hainfellner, "Selective darkening of the windshield using high speed video analysis for driving assistance," in *Proc. 13th Int. Conf. Optim. Elect. Electron. Equip.*, May 2012, pp. 1465–1470.
- [10] T. Taniguchi, K. Furusawa, H. Liu, Y. Tanaka, K. Takenaka, and T. Bando, "Determining utterance timing of a driving agent with double articulation analyzer," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 810–821, Mar. 2016.
- [11] H. Liu, T. Taniguchi, Y. Tanaka, K. Takenaka, and T. Bando, "Essential feature extraction of driving behavior using a deep learning method," in *Proc. IEEE Intell. Vehicles Symp.*, Jul. 2015, pp. 1054–1060.
- [12] L.-H. Zhao, Z.-Y. Liu, and H. Chen, "Design of a nonlinear observer for vehicle velocity estimation and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 664–672, May 2011.
- [13] H. Liu, T. Taniguchi, T. Takano, Y. Tanaka, K. Takenaka, and T. Bando, "Visualization of driving behavior using deep sparse autoencoder," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1427–1434.
- [14] B. N. Hilton, T. A. Horan, R. Burkhard, and B. Schooley, "SafeRoadMaps: Communication of location and density of traffic fatalities through spatial visualization and heat map analysis," *Inf. Vis.*, vol. 10, no. 1, pp. 82–96, 2011.
- [15] M. Treiber and D. Helbing, "Reconstructing the spatio-temporal traffic dynamics from stationary detector data," *Cooperat. Transp. Dyn.*, vol. 1, no. 3, pp. 1–24, 2002.
- [16] X. Huang, Y. Zhao, J. Yang, C. Zhang, C. Ma, and X. Ye, "Trajgraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 160–169, Jan. 2016.
- [17] M. Kilicarslan and J. Y. Zheng, "Visualizing driving video in temporal profile," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1263–1269.
- [18] K. Takeda *et al.*, "Self-coaching system based on recorded driving data: Learning from one's experiences," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1821–1831, Dec. 2012.
- [19] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
- [20] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, vol. 46. Hoboken, NJ, USA: Wiley, 2004.
- [21] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Artificial Neural Networks—ICANN*. Berlin, Germany: Springer, 1997, pp. 583–588.
- [22] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 194–281.
- [23] A. Ng, "Sparse autoencoder," CS294A Lecture Notes, Stanford Univ., Stanford, CA, USA, Tech. Rep. vol. 72, 2011, pp. 1–19.
- [24] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning—ICANN*. Berlin, Germany: Springer, 2011, pp. 52–59.
- [25] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 121–127, 2009.
- [26] A. D. Alvarez, F. S. Garcia, J. E. Naranjo, J. J. Anaya, and F. Jiménez, "Modeling the driving behavior of electric vehicles using smartphones and neural networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 3, pp. 44–53, Jul. 2014.
- [27] T. Tagawa, Y. Tadokoro, and T. Yairi, "Structured denoising autoencoder for fault detection and analysis," in *Proc. Asian Conf. Mach. Learn.*, 2014, pp. 96–111.
- [28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

- [29] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, nos. 4–5, pp. 411–430, Jun. 2000.
- [30] F. Wilcoxon, S. Katti, and R. A. Wilcox, "Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test," in *Selected Tables in Mathematical Statistics*. Providence, RI, USA: American Mathematical Society, vol. 1, pp. 171–259, Jun. 1970.



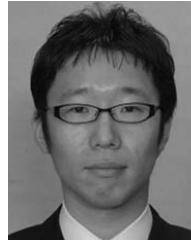
HaiLong Liu (S'15) received the B.E. and M.S. degrees from the College and Graduate School of Information Science and Engineering, Ritsumeikan University, in 2013 and 2015, respectively, where he is currently working toward the Ph.D. degree. He also has been a Research Fellow (DC) of the Japan Society for the Promotion of Science since 2016. His research interests include deep learning and its applications to driving behavior.



Tadahiro Taniguchi (M'13) received the M.E. and Ph.D. degrees from Kyoto University in 2003 and 2006, respectively. From 2008 to 2010, he was an Assistant Professor with the Department of Human and Computer Intelligence, Ritsumeikan University. Since 2010, he has been an Associate Professor with the Department of Human and Computer Intelligence, Ritsumeikan University. He is engaged in research into machine learning, emergent systems, and semiotics.



Yusuke Tanaka received the B.E. and M.S. degrees in engineering from Nagoya University in 2007 and 2009, respectively. He was with DENSO Corporation from 2009 to 2013. He is currently with Toyota Infotechnology Center Co., Ltd. His research interests include understanding and utilizing driving behavioral data and the applications of machine learning methods.



Kazuhito Takenaka (M'13) received the B.E. degree from The University of Tokyo in 2006, and the M.S. and Ph.D. degrees from the Graduate School of Information Science and Technology, The University of Tokyo, in 2008 and 2011, respectively. He is currently with DENSO Corporation. His research interests are understanding and utilizing driving behavioral data and the applications of machine learning methods.



Takashi Bando (M'13) received the B.E. degree in engineering from Kyoto Institute of Technology in 2002, and the M.E. and Ph.D. degrees from the Graduate School of Information Science, Nara Institute of Science and Technology in 2004 and 2007, respectively. He was with DENSO Corporation from 2007 to 2015. He is currently with DENSO International America, Inc. His research interests include understanding and realizing driving behavioral data and developing applications of machine learning methods for advanced driver assistance systems or automated driving systems.