IEICE TRANS. INF. & SYST., VOL.E103–D, NO.6 JUNE 2020

PAPER Special Section on Machine Vision and its Applications

Driver Drowsiness Estimation by Parallel Linked Time-Domain CNN with Novel Temporal Measures on Eye States

Kenta NISHIYUKI[†,††], **Jia-Yau SHIAU**[†††a)], **Shigenori NAGAE**[†], ***Nonmembers***, **Tomohiro YABUUCHI**[†], **Koichi KINOSHITA**[†], ***Members***, **Yuki HASEGAWA**[†], ***Nonmember***, **Takayoshi YAMASHITA**[††b)], ***and*** **Hironobu FUJIYOSHI**[††c)], ***Members***

SUMMARY Driver drowsiness estimation is one of the important tasks for preventing car accidents. Most of the approaches are binary classification that classify a driver is significantly drowsy or not. Multi-level drowsiness estimation, that detects not only significant drowsiness but also moderate drowsiness, is helpful to a safer and more comfortable car system. Existing approaches are mostly based on conventional temporal measures which extract temporal information related to eye states, and these measures mainly focus on detecting significant drowsiness for binary classification. For multi-level drowsiness estimation, we propose two temporal measures, average eye closed time (AECT) and soft percentage of eyelid closure (Soft PERCLOS). Existing approaches are also based on a time domain convolutional neural network (CNN) as deep neural network models, of which layers are linked sequentially. The network model extracts features mainly focusing on mono-temporal resolution. We found that features focusing on multi-temporal resolution are effective to multi-level drowsiness estimation, and we propose a parallel linked time-domain CNN to extract the multi-temporal features. We collected an own dataset in a real environment and evaluated the proposed methods with the dataset. Compared with existing temporal measures and network models, Our system outperforms the existing approaches on the dataset.

key words: *driver monitoring, driver drowsiness estimation, time-domain CNN, PERCLOS*

1. Introduction

Driver drowsiness is one of the leading causes of car accidents. Various approaches have been studied to construct an accurate estimator for driver drowsiness [1]–[23].

Most of the approaches are binary classification that classify whether a driver is significantly drowsy or not. It helps to avoid car accidents. However, even though a system detect a significantly drowsy driver correctly, it provides limited time to a car system until occurring accidents and the system can wake the driver only in an uncomfortable way such as a loud alert. On the contrary, multi-level drowsiness estimation, that detects not only "significant drowsiness" but also "moderate drowsiness", enables a car system feed appropriate intervention to a driver according to the drowsiness levels. For example, a system can recover comfortably a moderately drowsy driver with cold air. Zilberg et al. proposed a five-level drowsiness definition [24] that has been widely used [8], [11], [12], [22] for multi-level drowsiness estimation. We modify the five-level definition, and develop multi-level drowsiness estimation system following the modified definition.

Existing approaches show that conventional temporal measures, such as percentage of eyelid closure (PERCLOS) and blink frequency, are helpful to extract temporal information related to eye states for drowsiness estimation. These measures are designed to mainly focus on detecting drowsiness of high levels such as significantly drowsy and extremely drowsy. For detecting drowsiness of low levels such as moderately drowsy, we propose two temporal measures: average eye closed time (AECT) and soft percentage of eyelid closure (Soft PERCLOS). AECT is the average number of frames with eye closed in a blink interval. It is helpful to distinguish between a slightly drowsy driver who blinks frequently and a significantly drowsy driver who closes eyes for a certain time. Soft PERCLOS is the ratio of the number of frames with the eyes not fully opened, and helpful to detect a moderately drowsy driver whose eyes are not fully opened.

Some researches proposed deep learning based approaches with a time domain convolutional neural network (CNN). The time domain CNN can extract temporal features, and the features are effective to drowsiness estimation. The layers of the time domain CNN decrease the size of feature maps progressively, and temporal resolution of the feature maps is also decreased. The layers are linked sequentially, therefore, these models are designed to extract features mainly focusing on mono-temporal resolution. Shih and Hsu [4] proposed a multistage spatial-temporal network (MSTN), of which convolutional layers are linked in parallel. The feature maps of each convolutional layer are concatenated with the parallel linked structure, and fully connected layers estimate driver drowsiness with the concatenated feature maps. Therefore, MSTN can extract features of multi-spatial resolution. These features are effective to drowsiness estimation, but we found that features focusing on multi-temporal resolution are more effective to multi-level drowsiness estimation. To extract features of multi-temporal resolution, we propose a parallel linked time-domain CNN. Furthermore, we visualize which of input features, that is fed into the network model, are

Manuscript received August 31, 2019.
Manuscript revised February 15, 2020.
Manuscript publicized April 10, 2020.

[†]The authors are with Vision Sensing Lab., OMRON Corporation, Kizugawa-shi, 619–0283 Japan.
[††]The authors are with Department of Information Engineering, Chubu University, Nagoya-shi, 487–8501 Japan.
[†††]The author is with Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan.
a) E-mail: jiayau.shiau@gmail.com
b) E-mail: takayoshi@isc.chubu.ac.jp
c) E-mail: fujiyoshi@isc.chubu.ac.jp
DOI: 10.1587/transinf.2019MVP0017

important for the estimation as a sensitivity map, and validate whether our proposed model extracts features of multi-temporal resolution or not.

Combining our proposed temporal measures and parallel linked time-domain CNN, we conduct a multi-level drowsiness estimation system. First, the system detects a driver's face and eyes, and extracts features related to eyes on each driver image. The features include width, height of eyes, eye states which mean eye opening degree, etc. As a next step, the system calculates four temporal measures from sequences of the eye states to extract temporal information for drowsiness estimation. We utilize not only our proposed measures but also conventional measures to estimate drowsiness of various levels. Finally, the system estimates multi-level driver drowsiness with a parallel linked time-domain CNN.

Most researchers evaluate methods with datasets that are recorded in a driving simulator. The condition such as a background, vibration, and illumination is different from that of a real environment. Therefore, we collect an own dataset that is recorded in a real environment. To verify the effectiveness of the proposed system, we conduct experiments with the dataset. The experiments have demonstrated that the proposed temporal measures and parallel linked time-domain CNN outperforms conventional temporal measures and network models. We also visualize the prediction results as line graphs, show the correlation between the prediction results and groundtruth. For showing possibility of early detection of driver drowsiness, we evaluate the proposed system on each drowsiness level, and investigate the transition time from drowsiness of low levels to high levels. The results show that the proposed system achieves high accuracy on moderate drowsiness, and the transition time from moderate drowsiness to significant drowsiness is sufficiently long for recovering comfortably a drowsy driver. As a result, our proposed system can estimate a drowsy driver early.

Our contributions are summarized as follows:

- We propose a multi-level driver drowsiness estimation system. The system consists of three major components: (1) calculating features related to eyes from each driver image, (2) calculating temporal measures on eye states, and (3) estimating drowsiness levels with time-domain convolutional neural network (CNN). (Sect. 3)
- On the second component, we propose novel temporal measures for detecting low level drowsiness: AECT and Soft PERCLOS. (Sect. 3.2) On the third component, we propose a parallel linked time-domain CNN to extract features focusing on multi-temporal resolution. (Sect. 3.4)
- We evaluate our proposed method with a driving movie dataset recorded in a real environment. (Sect. 4)
- We validate that our proposed method captures the change of driver drowsiness with line graphs and extract features focusing on multi-temporal resolution with sensitivity maps that is generated by SmoothGrad [25]. (Sect. 4.4)
- We also show experimental results related to early detection of a drowsy driver. (Sect. 4.5)

2. Related Work

We show categories focusing on sensors to capture information of a drowsy driver in this section. Then, we discuss the most relevant approaches for accurate multi-level driver drowsiness estimation.

Approaches to Estimate Driver Drowsiness: The approaches to estimate driver drowsiness are divided into three categories: driving patterns of cars, physiological features of drivers, and visual expressions of faces.

The first category is based on driving patterns such as steering wheel movements, braking time series, and lane departure [13]–[15]. These approaches are user-friendly but are influenced by other factors unrelated to drowsiness such as driving skills, road conditions, and car characteristics.

The second category uses electrical bio-signals from electroencephalograms (EEG) [16]–[21], electrocardiograms (ECG) [22], [23], and electrooculograms (EOG) [22], [23]. These approaches are accurate but uncomfortable for the driver.

The third category focuses on analyzing sequences of driver images to extract facial appearances such as eye closure, head movement, yawning, eye focus, and comprehensive facial expressions [1]–[6]. These approaches are user-friendly, as accurate as the other approaches, and less influenced by factors unrelated to drowsiness than driving-pattern based approaches. Therefore, we use facial appearances to estimate driver drowsiness.

Temporal Measures related to eyes: Some researchers proposed methods to estimate driver drowsiness with handcrafted temporal measures related to eyes. Eye closure is the most commonly used to estimate drowsiness. Percentage of eyelid closure (PERCLOS) [7] and blink frequency [26] are also widely used [8], [9]. Wierwille et al. mentioned that PERCLOS strongly correlates with driver drowsiness [7]. PERCLOS measures the ratio of time with eyes closed to a given time. Zhang et al. mentioned that blink frequency is also an important measure for drowsiness estimation [26]. The definition of blink frequency is as follows:

$$f^t_{blink} = \frac{n^t_{blinking}}{N^t_{total}}, \tag{1}$$

where N^t_{total} denotes the total number of frames, and $n^t_{blinking}$ denotes the number of frames that the eye state changes between open and closed for a given period of time t. The definition of PERCLOS is as follows:

$$PERCLOS^t = \frac{n^t_{close}}{N^t_{total}}, \tag{2}$$

where n^t_{close} denote the number of frames with eyes closed for a given period of time t.

These temporal measures are designed for binary classification. Hence, we propose novel temporal measures for

multi-level drowsiness estimation.

CNN Based Drowsiness Estimation: Convolutional neural network (CNN) is used to extract features directly from driver facial images. Lyu et al. [10] proposed binary classification with a CNN and long short-term memory (LSTM) [27], [28]. Focusing on eye and month states, Reddy et al. [6] proposed drowsiness estimation of three levels (drowsy, yawning, and normal) with a CNN. Huynh et al. [5] utilized temporal information of movies with a 3D-CNN [29]. Shih and Hsu [4] proposed MSTN for binary classification with a CNN (VGG-16 [30]) and LSTM. MSTN can learn information of multi-spatial resolutions. For multi-level drowsiness estimation, we consider that information of multi-temporal resolutions is more effective than multi-spatial resolutions.

Approaches of Multi-level Drowsiness Estimation: The approaches with a CNN achieved high accuracy, but they are all binary classification and not suitable for preventing drowsy driving. Some researchers have tried to prevent drowsy driving with multi-level drowsiness levels. From sequences of face images, Nakamura et al. [11] extracted hand-crafted features such as eyelid movements and wrinkle changes, and estimated five levels of drowsiness with the traditional k-NN method. Sun et al. [12] used only sequences of eye-blinks to estimate five levels of drowsiness with deep neural networks that are simple 1D time-domain convolution or LSTM. The time-domain convolution with eye blinks is effective, but we consider that time-domain convolution with temporal measures that are designed for multi-level drowsiness estimation is more effective. Both conventional approaches are evaluated with datasets that are recorded in a driving simulator.

3. Method

Our system consists of three components: (1) calculating features related to eyes from each driver image, (2) calculating temporal measures on eye states, and (3) estimating drowsiness levels with a parallel linked time-domain CNN. The details of the system are shown in Fig. 1.

3.1 Drowsiness Level

We use the definition of the drowsiness levels proposed by Zilberg et al. [24]. They classify drowsiness into five levels: (1) alert, (2) slightly drowsy, (3) moderately drowsy, (4) significantly drowsy, and (5) extremely drowsy. In our experiment, we simplify the scale of the rating from the five to four levels by merging the first two levels into one level. If a system can detect a moderately drowsy driver, the system can have a sufficiently long time to provide appropriate intervention to the driver. This is because the difference between alert and slightly drowsy is not important, but distinguishing between alert and moderately drowsy is important for an car system. The four levels are: (1) alert, (2) moderate drowsy, (3) significantly drowsy, and (4) extremely drowsy.

3.2 Calculating Features from Each Driver Image

First, we extract some features from each driver image. The features consist of (1) center positions of both eyes and pupils, (2) average widths and heights of eyes, and (3) eye

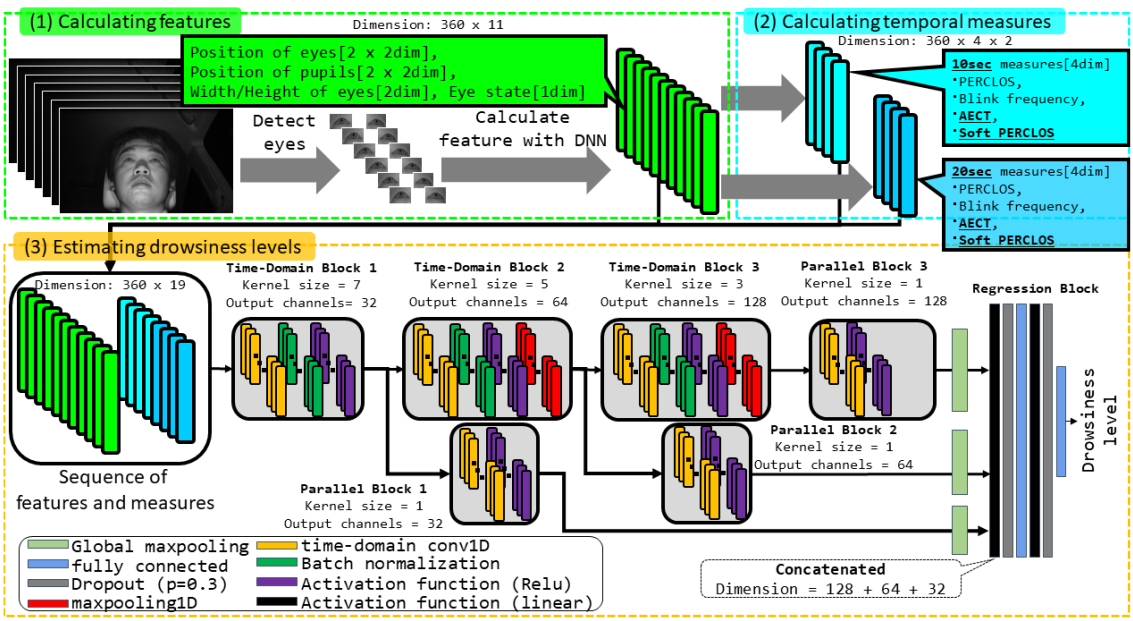

Fig. 1 The our system architecture: (1) The system detects eyes of the driver and calculates features related to eyes from each driver image. (2) Using the features, the system calculates temporal measures on eye states with time periods of 10 and 20 seconds: PERCLOS, Blink frequency, AECT, Soft PERCLOS. (3) Using the features and temporal measures, the system estimates drowsiness levels of the driver with time-domain convolution including a parallel linked structure.

state. The eye positions are determined by using OKAO Vision [31], then the eyes images are clipped and resized. The eyes images are fed into a ResNet [32] to estimate the above features. The ResNet has three residual blocks and 13 layers, and receives an eye image of 64×64. The eye state ranges from -1.0 to 1.0. If the state is greater than 0.0, the eye is open. To train the ResNet, we annotated 24 points around an eye and eye states on eyes images. The ResNet estimates the 24 points and eye states, and extracts the features from points and eye states.

3.3 Calculating Temporal Measures on Eye States

We calculate temporal measures on the eye status from the sequence of the eye states that are estimated with the ResNet based CNN.

PERCLOS and blink frequency are widely used and known as effective temporal measures to denote eye states for binary classification. In addition to the measures, we propose novel temporal measures for multi-level drowsiness estimation: AECT and Soft PERCLOS.

AECT (average eye closed time): We propose a novel measure named AECT, which is the average number of frames with eye closed in a blink interval. We observe that significantly drowsy drivers tend to close their eyes for a long time, therefore, we add AECT as a feature. Although it slightly overlaps with PERCLOS, we found that AECT improves the accuracy. The definition of AECT is as follows:

$$AECT^t = \frac{PERCLOS^t}{f^t_{blink}}. \tag{3}$$

AECT is similar to average eyes closed speed (AECS) [1] but slightly different. Because AECS measures the speed of blinking eyelids, it requires a very high frame-rate camera. In contrast, AECT can be used with a low frame-rate (30 frames per second (FPS)) camera.

Soft PERCLOS: We propose a novel measure named 'Soft PERCLOS' for drowsiness estimation. Soft PERCLOS denotes the ratio of the number of frames with the eyes not fully open. This is a strong hint to detect a moderately drowsy state. The definition of soft PERCLOS is as follows:

$$PERCLOS^t_{soft} = \frac{n^t_{soft\_close}}{N^t_{total}}, \tag{4}$$

where $n^t_{soft\_close}$ is the number of frames with $S_{eye} < 0.8$ for a given time t. When a driver is moderately drowsy with eyes not fully closed, the driver may blink quickly with the eye state dangling around 0. In that case, it is difficult to detect drowsy driving with only PERCLOS, blink frequency, and AECT. However, we found that we can estimate it by adding Soft-PERCLOS.

Multiscale Time Period: As shown in Eq. (1), (2), (3), and (4), the temporal measures require a time period t. We found that the time period affects the accuracy and multiscale time period improve the accuracy. The "multiscale time period" indicates using multiple temporal measures that are calculated by different time periods. Our proposed parallel linked time domain CNN extracts features of multi-temporal resolution, therefore, the multiscale time period are slightly overlapped with the features. However, we found that combining the parallel linked time-domain CNN with the multiscale time period improves the accuracy. In our experiments, we use time periods of 10 and 20 seconds.

We concatenate the features calculated from each driver image and the multiscale results of the measures. They are calculated from the frames that are determined by two parameters: specific time period T and frames per second fps.

3.4 Estimating Drowsiness Levels with Parallel Linked Time Domain CNN

Our proposed network model consists of three parts: (1) time-domain convolution block, (2) paralleled smooth block, and (3) regression block. The model receives a sequence of the features and measures described in Sect. 3.2 and Sect. 3.3. The features are center positions of both eyes and pupils, width and heights of eyes, and eye state. The measures are PERCLOS, blink frequency, AECT, and Soft-PERCLOS. The model outputs the drowsiness level of the last frame. The architecture is shown in Fig. 1.

3.4.1 Time-Domain Convolution Block

We apply time-domain convolution on sequences of the features and measures to extract temporal information. We use three time-domain convolution blocks, with kernel size set to 7, 5, and 3 respectively. The three time-domain convolution blocks are linked sequentially. The second and third blocks have max polling layers. Note that different input features are treated as feature channels like the RGB channels of images.

3.4.2 Paralleled Smooth Block

The paralleled smooth blocks are used to extract deep features focusing on multi-temporal resolutions from the time-domain convolution blocks. The parallel linked structure is inspired by MSTN [4], where it is used for not time-domain convolution but usual convolution in the spatial or feature domain. Therefore, MSTN is the structure to extract features focusing on multi-spatial resolution. On the other hand, we link the parallel smooth blocks after each time-domain convolution block to extract features focusing on multi-temporal resolution. The number of convolution kernels of parallel blocks is the same as input channels.

3.4.3 Regression Block

We use global max pooling layers to concatenate features of different parallel smooth blocks. The outputs of parallel smooth blocks will be one dimension after the global max

Table 1 Variance of groundtruths: "variance" means the average of variances that is calculated from drowsiness scores on each frame. "max - min" means the average of differences between maximum and minimum scores on each frame. "ratio ([max - min] > threshold)" means ratio of the frames, the difference of which between maximum and minimum scores is greater than the threshold.

Item	Value
variance	0.1942
max - min	0.6753
ratio ([max - min] > 0.5)	53.99%
ratio ([max - min] > 1.0)	12.88%
ratio ([max - min] > 2.0)	2.53%

pooling layers. The deep features of multi-temporal resolution after global max pooling are concatenated and fed into the regression block. For the regression block, we simply use two fully connected layers with linear activation functions to estimate drowsiness levels. Note that the last block except for the global max pooling layers cannot be combined into a simple projection layer, because a dropout operation is carried out after the first linear activation when training.

4. Experiments

We present experimental results under different conditions. First, the details of the dataset will be introduced in Sect. 4.1. We explain our experimental details in Sect. 4.2. We show the experimental results with different model architectures and input features in Sect. 4.3. We also show the experimental results to evaluate that our system catches the change of driver drowsiness and extract features focusing on multi-temporal resolution with sensitivity maps that is generated by SmoothGrad [25] in Sect. 4.4. Finally, we mention additional experiments for early detection of drowsiness in Sect. 4.5.

4.1 Dataset

Our dataset is recorded in a real driving car. For safety reasons, we attached an IR camera in front of the front seat passenger in real cars. We instructed subjects to look ahead with a feeling of driving. To make the datasets diversified, we employed 16 different subjects, some of whom wore glasses or masks. Each subject was recorded for around 30 minutes with an IR camera, 60 FPS. For the annotation, the videos were cut into 5-second clips, and three workers annotated the clips individually. Consequently, each worker labeled around 360 clips for each of the 16 subjects. Finally, we used the average of the annotation as a groundtruth and interpolated the groundtruth with linear interpolation in the time domain. Therefore, we can get the interpolated drowsiness level on each frame, and use it as a groundtruth. Note that, we instructed the workers to train adequately how to annotate precise drowsiness scores. As a result, variance of the drowsiness scores is low on our dataset. The details of the variance are shown in Table 1.

Examples of the dataset are shown in Fig. 2. These examples demonstrates the meticulousness of our experimental environment and dataset.

Fig. 2 One of the subjects in our dataset: This figure shows three cases of different drowsiness levels with four sequential cropped images for each case. The time distance of each two sequential images is 0.1 seconds approximately.

4.2 Experimental Details

Input: Our system receives 30 seconds image sequences of 12 FPS. When we use 12 FPS, we can detect blinks of the driver, and accelerate the system. For each frame, the system estimates coordinate of centers of eyes and pupils, average width and height of eyes, eye state, and multi-scale temporal measures: $PERCLOS$, f_{blink}, $AECT$, and $PERCLOS_{soft}$ with $t = (10sec, 20sec)$. Our network model in the system receives their feature sequences. Their dimensions are 8, 2, 1, and 8, respectively. To be precise, the shape of the model input is 360×19 shown as Fig. 1 (3). 360 means the number of input frames, and 19 means the dimension number of input features. Note that, the system calculate the features and temporal measures with a sliding-window style for each of the continuous 360 frames. The temporal measures are calculated from the frames of 10sec and 20sec length, and the frames are overlapping. Therefore, the measures might include some of redundant computation. However, for sake of simplicity, we complete the dimensional numbers of the features and temporal measures. Meanwhile, we also evaluate some network models the inputs of which are eye image sequences. The images are resized to 64×32, and the shape of the model input is 360×2048.

Hyperparameters of network: We select Adam with hyperparameters set as follows: $lr = 0.001$, $betas = (0.9, 0.999)$, $eps = 1e^{-8}$, $weight\ decay = 0.0005$. We use the L1 loss, also known as mean absolute error (MAE).

Training and testing: We evaluate our model with a leave-one-out cross validation. The dataset is split by subjects, one is used for testing, and the others are used for training. We train the model with 1,024 instances on each epoch. The instance indicates 30 seconds clip and input of the network model. The instances randomly picked out from the training dataset that includes 15 subjects on each epoch. The number of training epoch is 100, the total number of instances is 102,400. The shape of the instance is 360×19. Meanwhile,

we divide the testing dataset into every instance, use all of their instances for testing. We use a batch size of 128 when inputs of the network are sequences of features. If sequences of eye images are fed into a model, the batch size is 4 for the limitations of GPU memory. In the testing, we apply exponential moving average (EMA) on both predictions and groundtruth for denoising. The window length of EMA is 30 frames.

Metrics: The accuracy is calculated by the number of correct predictions over the total test dataset. A correct prediction is defined as follows:

$$Correct = \begin{cases} 1, \ if \ |Y_i - \hat{Y}_i| < M, \\ 0, \ otherwise. \end{cases}$$

where Y_i is the prediction of ith frame, $\hat{Y}_i$ is the groundtruth, and M is the width of the accurate margin (tolerance). We also evaluate with MAE.

Model architecture: As noted in Sect. 3.4, Our proposed network model is a parallel linked time-domain CNN. For comparison, we also evaluate seven model architectures: LSTM, VGG-LSTM, VGG-LSTM with parallel linked structure, VGG-LSTM with time-domain pooling, 3D-CNN, 1 time-domain convolution block, and 3 time-domain convolution blocks.

The details of the model architectures we used in our experiments are shown in Tables 2, and 3. Table 2 indicates the models that receive sequences of eye images. The eye images are resized to 64×32. Table 3 indicates the models that receive sequences of the features as noted in Sect. 3.

The time-domain pooling model is designed following the idea of learning temporary information of Ng et al. [33]. Features are calculated from sequences of eye images with VGG, and the features are downsampled with max pooling in the time domain. In the time-domain pooling, we estimate drowsiness with the same layers as VGG-LSTM. The 1 time-domain convolution block is the model with only one time-domain convolution block and linear regression (fully connected layer). The 3 time-domain convolution blocks is the model with three time-domain convolution blocks linked sequentially. The final one is our proposed model as described in Sect. 3.

Table 2 Architectures of models that receive sequences of eye images: On "lstm", and "fc", the argument is the hidden size. "do" is a dropout layer: the argument is the probability. "bn" is a batch normalization layer. "c" is a convolutional layer, and "c3" is a 3D convolutional layer: the first argument is the number of the output channels, and the second is a kernel size. "mp" is a max pooling layer, and the argument is the kernel size. "block" is the block of VGG as in [30]: the first argument is the output channels, the second is a number of convolutional layers.

VGG-LSTM	VGG-LSTM (parallel)			3D-CNN
block(64,2), mp(2)	block(64,2), mp(2)			c3(16,7), bn, relu
block(128,2), mp(2)	block(128,2), mp(2)			c3(32,5), bn, relu
block(256,2), mp(2)	block(256,2), mp(2)		c(128,1), relu	mp(2)
block(512,2), mp(2)	block(512,2), mp(2)	c(256,1), relu	-	c3(64,3), bn, relu
block(512,2), mp(2)	block(512,2), mp(2)	-	-	-
-	c(512,1), relu	-	-	-
global MP				
-	concat			-
bn, do(0.3), fc(128), bn, relu, do(0.4), fc(128), relu				bn, do(0.3), fc(64)
lstm(64), bn, fc(1)				do(0.3), fc(1)

Table 3 Architectures of models that receive sequences of features.

LSTM	1 time domain conv block	3 time domain conv blocks	parallel linked time domain CNN (3 time-domain convolution blocks, ours)		
lstm(256) do(0.05)	c(16,3) bn, relu	c(32,7) bn, relu	c(32,7) bn, relu		
lstm(256) do(0.05)	-	c(64,5) bn, relu mp(2)	c(64,5) bn, relu mp(2)		c(32,1), relu
lstm(256) do(0.05)	-	c(128,3) bn, relu mp(2)	c(128,3) bn, relu mp(2)	c(64,1), relu	-
-	-	-	c(128,1), relu	-	-
-	global mp				
-	-	-	concat		
bn, do(0.3), fc(64), do(0.3), fc(1)					

4.3 Performance of Proposed Method

We perform two experiments: (1) experiments with different model architectures, and (2) experiments with different input features. We perform the experiments with different model architectures to validate the effectiveness of our proposed network architecture that is a parallel linked time domain CNN. We perform the experiments with different input features to validate the effectiveness of our proposed temporal measures: AECT and Soft PERCLOS.

Different model architectures: We performed the cross validation on different model architectures with the same input. The accuracy of different models is shown in Table 4. Our proposed model performed better than the other models. The bottom four models that receive sequences of features performed better than the top four models that receive sequences of eye images. On our dataset, it indicates that the hand-crafted features are better than features extracted automatically from eye images with the CNNs such as VGG and 3D-CNN. CNNs need enormous dataset to extract features automatically, but collecting real driver drowsiness dataset is difficult. We found that our proposed hand-crafted features improve the accuracy on limited datasets such as driver drowsiness. Among the models that receives sequences of features, Our proposed model achieves the best accuracy. It indicates that our parallel linked time domain CNN is ef-

Table 4 Experimental results for different model architectures: As mentioned in Sect. 4.2, we calculated the accuracy and MAE. The accuracy is calculated with M (width of accurate margin) of 1.0 and 0.5. The input of the top four models is sequences of eye images. The input of the bottom four models is sequences of features.

Model	M=1.0	M=0.5	MAE
VGG-LSTM	73.89%	37.54%	0.6971
VGG-LSTM (parallel)	66.58%	31.64%	0.7990
VGG-LSTM (time-domain pool)	55.50%	28.41%	0.9695
VGG-LSTM (time-domain pool, parallel)	48.20%	21.20%	1.3065
3D-CNN	69.31%	32.02%	0.7535
LSTM	60.71%	37.45%	1.1115
1 time-domain convolution block	80.97%	54.16%	0.5428
3 time-domain convolution blocks	**97.99%**	66.99%	0.3931
parallel linked time domain CNN (3 time-domain convolution blocks, ours)	96.79%	**69.04%**	**0.3785**

Table 5 Experimental results for different input features: The top two features are calculated on frame-by-frame basis. The following two features are temporal measures on eye states. We adapt multiscale time period on the last features.

Features (dimension)	M=1.0	M=0.5	MAE
Displacement (8)	95.03%	56.72%	0.4737
+ Width, height and eye state (11)	95.53%	68.03%	0.4061
+ Conventional temporal measures: PERCLOS and blink frequency (13)	96.27%	66.25%	0.3996
+ **Proposed temporal measures: AECT and Soft PERCLOS (15)**	**96.90%**	67.98%	0.3971
+ **Multiscale time period (19)**	96.79%	**69.04%**	**0.3785**

fective to learn temporal information for driver drowsiness estimation.

Different input features: We also performed experiments on different input features with our proposed model. The results are shown in Table 5. From the top to the bottom, we add features that is fed into our proposed model. The top two features are calculated on frame-by-frame basis. The displacement features consist of the center position of both eyes and pupils. We add the average width and height of left and right eyes, and the eye state in the second row.

The following two features are temporal measures on eye states. In the third row, we add the conventional temporal measures: PERCLOS, and blink frequency. In the fourth row, we add our proposed temporal measures: AECT, and Soft PERCLOS. Note that, these temporal measures are calculated with a time period of 20 seconds (t = 20).

The last feature is calculated with multiscale time period as noted in Sect. 3.3. We add the temporal measures calculated with a time period of 10 seconds (t = 10) in the last row.

Our model performs well even if only displacement features are given. The temporal measures on eye states is effective for driver drowsiness estimation. Our proposed temporal measures, AECT and Soft PERCLOS, perform better than with only conventional temporal measures. Furthermore, the multiscale time period also improves the accuracy.

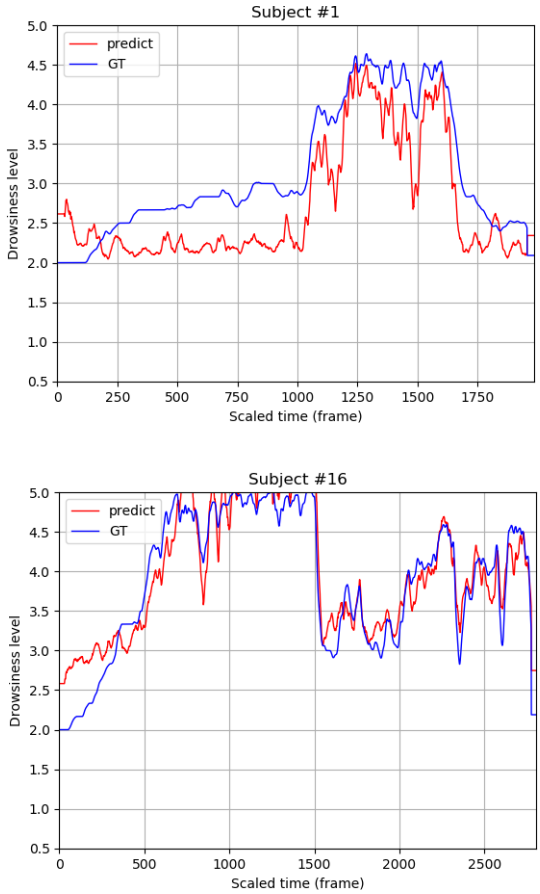

Fig. 3 Prediction results: The x axis denotes time (or frame) and y axis denotes the drowsiness level. The blue line is the groundtruth, the red line is the prediction results. We apply exponential moving average (EMA) on both predictions and groundtruth to make the visualized results smooth.

4.4 Analysis

In this subsection we show the experimental results to analyze our proposed method. We perform two experiments: (1) experiments of visualization of prediction results, and (2) experiments of visualization of sensitivity maps. We visualize the prediction results with line graphs, and check subjectively whether our system can capture the change of driver drowsiness or not. We also visualize the sensitivity maps, that is generated by SmoothGrad [25], to validate whether the parallel linked time domain CNN extract features focusing on multi-temporal resolution or not.

Visualization of Prediction results: The line graphs of predictions and groundtruths are shown in Fig. 3. The predictions (red) of our proposed model fit the groundtruth, and their trends are correlated. The figure indicates that the time lag between predictions and groundtruth is small, and hence our system can catch the change of driver drowsiness well.

Visualization of Sensitivity Maps: We visualize the sensitivity maps, that is associated to input features, with SmoothGrad [25]. The sensitivity maps are shown in Fig. 4. We visualize sensitivity of two models: our proposed time

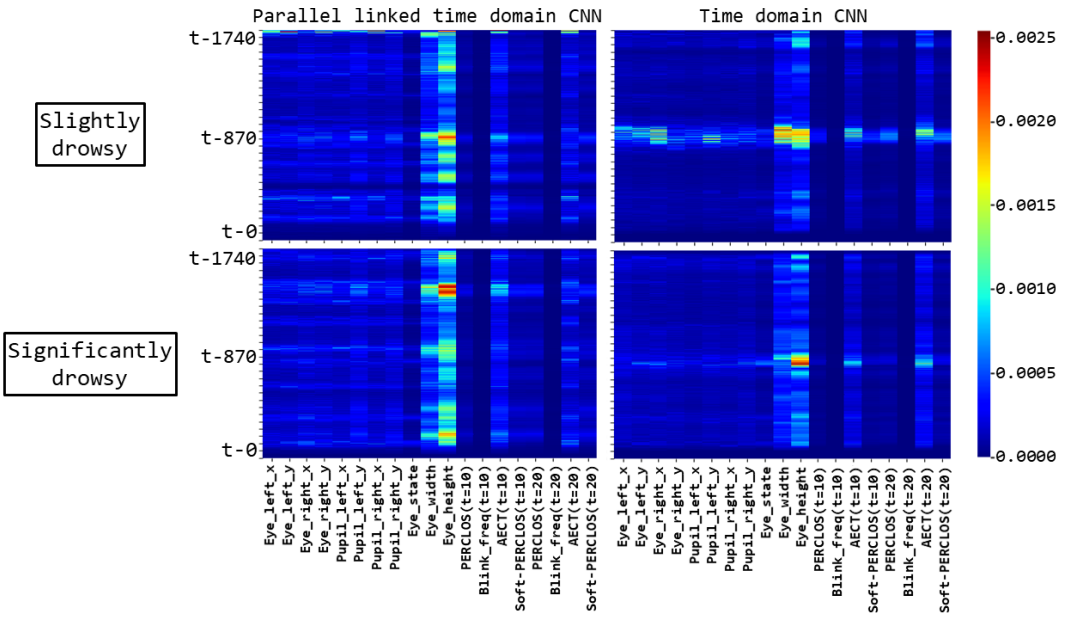

Fig. 4 Visualization of sensitivity maps: The map shows that the sensitivity to input features. The cool and warm colors are associated to low and high sensitivity, respectively. The column denotes features: center positions of both eyes and pupils, average widths and heights of eyes, eye state, and temporal measures from left to right. The row denotes frames: (t - 1740)-th frame, (t - 1680)-th frame, ..., t-th frame from top to bottom. The top and bottom maps are 'slightly drowsy' and 'significantly drowsy', respectively. The left and right maps are our proposed method, that is parallel linked time domain CNN, and conventional time domain CNN, respectively.

domain CNN with three time domain convolution blocks linked in parallel, and a conventional time domain CNN with the blocks linked sequentially.

The center position of both eyes and pupils affect the prediction results moderately. The eye state affects the result slightly, and the width and height of eyes affects significantly. The real width is hardly changed regardless of driver drowsiness. However, the width, that is predicted by our system, is strongly correlated with the eye state and height of eyes. Both of the models leverage the width instead of the eyes state. Among temporal measures, our system utilizes the AECT instead of the blink frequency, and both of PERCLOS and Soft-PERCLOS affects the results moderately. We utilize multiscale time period which are t of 10 and 20 seconds, both of them affect the result.

The time domain CNN utilize features of a limited number of frames. Meanwhile, our proposed parallel linked time domain CNN utilize features of a variety of frames. The difference between the time domain CNN and the parallel linked time domain CNN can be confirmed at both of 'slightly drowsy' and 'significantly drowsy'. Evidently, our proposed model extract features focusing on multi-temporal resolution.

4.5 Early Detection of Drowsiness for Preventing Drowsy Driver

Drowsiness of high levels such as "significantly drowsy" and "extremely drowsy" cause serious accidents. If a system can estimate drowsiness of a low level such as "moderately drowsy" and transition time from "moderately drowsy" to "significantly drowsy" is sufficiently long, the system can provide more options to avoid accidents about drowsy driver.

Table 6 Experimental results on each drowsiness level: We calculated the accuracy, that is calculated with M (tolerance) of 1.0 and 0.5, and MAE. The accuracy is calculated by our proposed system. As mentioned in Sect. 3.1, we merged "alert" and "slightly drowsy" into one level.

Drowsiness level	M=1.0	M=0.5	MAE
alert, slightly drowsy	90.58%	54.71%	0.4900
moderately drowsy	**99.12%**	**74.71%**	**0.3847**
significantly drowsy	91.48%	51.80%	0.5460
extremely drowsy	96.82%	45.12%	0.5030

In this subsection, we show the experimental results for early detection of drowsiness. We perform two experiments: (1) experiments of evaluation on each drowsiness level, and (2) experiments of the transition time from drowsiness of lower levels to higher levels.

Evaluation on each drowsiness level: We evaluated our proposed method on each drowsiness level. The accuracy on each level is shown in Table 6. Our proposed method performs well on not only "extremely drowsy" but also "moderately drowsy". As the result, our system can detect drowsy driver in an early stage. The accuracy of "significantly drowsy" is worse than the others. The driver of "significantly drowsy" sometimes rubs his or her eyes and yawn. In those cases, the height of eyes is excessively small, therefore, our system sometimes misclassifies the driver as "ex-

Table 7 Transition time on each levels: We investigate the transition time from lower levels to higher levels.

Drowsiness level	Average [*sec*]	Max [*sec*]	Min [*sec*]
From "moderately drowsy" to "significantly drowsy"	**375.93**	**768.06**	**113.3**
From "significantly drowsy" to "extremely drowsy"	1299.34	4830.10	135.65

tremely drowsy". For preventing the misclassification, other features besides them related to eyes could be considered as helpful.

Transition Time: We investigate the transition time from lower drowsiness levels to higher levels. The transition time is shown in Table 7. The table show that from lower levels to higher levels is longer than 110 seconds.

Awaking the driver who is extremely drowsy with weak intervention is hard. Meanwhile, a system can awake easily a driver of "moderately drowsy" with weak intervention. Our system can detect preciously driver of "moderately drowsy", and the transition time from lower levels to higher levels is longer than about two minutes. As the result, our proposed method enables a system to recover the driver comfortably with weak intervention such as cold wind.

5. Conclusions

This paper presented a vision-based driver drowsiness estimation system from sequences of driver images. We proposed the novel model architecture with time-domain convolution including a parallel linked structure, and temporal measures: AECT, and Soft PERCLOS. We show that our proposed methods are effective for driver drowsiness estimation, on the two experiments: with different model architectures, and different input features. Our system predicts drowsiness levels with an overall accuracy of 96.79% and 69.04% with an error-tolerant value of 1 and 0.5, respectively. Moreover, its MAE is 0.3785%.

We show that our proposed parallel linked time domain CNN can extract features focusing on multi-temporal resolution with sensitivity maps. Furthermore, we also show possibility of early detection of drowsiness for preventing drowsy driver.

Future research should consider the potential effects of driver's actions, for example controlling a real driving car. Other datasets, that are recorded in a driving simulator, can validate the effects. Therefore, evaluating with the simulator dataset could be a complemental experiment.

References

[1] L. Lang and H. Qi, "The study of driver fatigue monitor algorithm combined PERCLOS and AECS," International Conference on Computer Science and Software Engineering (CASCON), pp.349–352, 2008.

[2] M. Omidyeganeh, A. Javadtalab, and S. Shirmohammadi, "Intelligent driver drowsiness detection through fusion of yawning and eye closure," IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), pp.1–6, 2011.

[3] F. Zhang, J. Su, L. Geng, and Z. Xiao, "Driver fatigue detection based on eye state recognition," International Conference on Machine Vision and Information Technology (CMVIT), pp.105–110, 2017.

[4] T.-H. Shih and C.-T. Hsu, "MSTN: multistage spatial-temporal network for driver drowsiness detection," Asian Conference on Computer Vision (ACCV) Workshops, vol.10118, pp.146–153, 2016.

[5] X.-P. Huynh, S.-M. Park, and Y.-G. Kim, "Detection of driver drowsiness using 3D deep neural network and semi-supervised gradient boosting machine," Asian Conference on Computer Vision (ACCV) Workshops, vol.10118, pp.134–145, 2016.

[6] B. Reddy, Y.-H. Kim, S. Yun, C. Seo, and J. Jang, "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks," IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp.438–445, 2017.

[7] W.W. Wierwille, S.S. Wreggit, C. Kirn, L.A. Ellsworth, and R.J. Fairbanks, "Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness. final report," National Highway Traffic Safety Administration, no.DOT HS 808 247, 1994.

[8] M. Tsujikawa, Y. Onishi, Y. Kiuchi, T. Ogatsu, A. Nishino, and S. Hashimoto, "Drowsiness estimation from low-frame-rate facial videos using eyelid variability features," International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp.5203–5206, 2018.

[9] J.W. Baek, B.-G. Han, K.-J. Kim, Y.-S. Chung, and S.-I. Lee, "Real-time drowsiness detection algorithm for driver state monitoring systems," International Conference on Ubiquitous and Future Networks (ICUFN), pp.73–75, 2018.

[10] J. Lyu, Z. Yuan, and D. Chen, "Long-term multi-granularity deep framework for driver drowsiness detection," arXiv preprint arXiv:1801.02325, 2018.

[11] T. Nakamura, A. Maejima, and S. Morishima, "Driver drowsiness estimation from facial expression features computer vision feature investigation using a cg model," International Conference on Computer Vision Theory and Applications (VISAPP), pp.207–214, 2014.

[12] M. Sun, M. Tsujikawa, Y. Onishi, X. Ma, A. Nishino, and S. Hashimoto, "A neural-network-based investigation of eye-related movements for accurate drowsiness estimation," International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp.5207–5210, 2018.

[13] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, "Steering wheel behavior based estimation of fatigue," International Driving Symposium on Human Factors in Driving Assessment, Training and Vehicle Design, pp.118–124, 2009.

[14] H. Malik, F. Naeem, Z. Zuberi, and R. ul Haq, "Vision based driving simulation," International Conference on Cyberworlds (CW), pp.255–259, 2004.

[15] R.F. Knipling and W.W. Wierwille, "Vehicle-based drowsy driver detection: Current status and future prospects," The Intelligent Vehicle-Highway Society of America (IVHS America), 1994.

[16] Z. Mardi, S.N.M. Ashtiani, and M. Mikaili, "Eeg-based drowsiness detection for safe driving using chaotic features and statistical tests," Journal of medical signals and sensors, vol.1, no.2, pp.130–137, 2011.

[17] M.V.M. Yeo, X. Li, K. Shen, and E.P.V. Wilder-Smith, "Can SVM be used for automatic EEG detection of drowsiness during car driving?," Safety Science, vol.47, no.1, pp.115–124, 2009.

[18] C.-T. Lin, C.-J. Chang, B.-S. Lin, S.-H. Hung, C.-F. Chao, and I.-J. Wang, "A real-time wireless brain–computer interface system for drowsiness detection," IEEE Transactions on Biomedical Circuits and Systems, vol.4, no.4, pp.214–222, 2010.

[19] C.-T. Lin, L.-W. Ko, I.-F. Chung, T.-Y. Huang, Y.-C. Chen, T.-P. Jung, and S.-F. Liang, "Adaptive eeg-based alertness estimation

system by using ica-based fuzzy neural networks," IEEE Transactions on Circuits and Systems I: Regular Papers, vol.53, no.11, pp.2469–2476, 2006.
[20] A. Picot, S. Charbonnier, and A. Caplier, "Drowsiness detection based on visual signs: blinking analysis based on high frame rate video," IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp.801–804, 2010.
[21] H. Albalawi and X. Li, "Single-channel real-time drowsiness detection based on electroencephalography," International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp.98–101, 2018.
[22] A. Tsuchida, M.S. Bhuiyan, and K. Oguri, "Estimation of drowsiness level based on eyelid closure and heart rate variability," International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp.2543–2546, 2009.
[23] A. Tsuchida, M.S. Bhuiyan, and K. Oguri, "Estimation of drivers' drowsiness level using a neural network based error correcting output coding method," International IEEE Conference on Intelligent Transportation Systems (ITSC), pp.1887–1892, 2010.
[24] E. Zilberg, Z.M. Xu, D. Burton, M. Karrar, and S. Lal, "Methodology and initial analysis results for development of non-invasive and hybrid driver drowsiness detection systems," International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless), p.16, 2007.
[25] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," arXiv preprint arXiv:1706.03825, 2017.
[26] W. Zhang, B. Cheng, and Y. Lin, "Driver drowsiness recognition based on computer vision technology," Tsinghua Science and Technology, vol.17, no.3, pp.354–362, 2012.
[27] F.A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Computation, vol.12, no.10, pp.2451–2471, 2000.
[28] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," arXiv preprint arXiv:1409.2329, 2014.
[29] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.35, no.1, pp.221–231, 2013.
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[31] "OKAO Vision." https://plus-sensing.omron.com/.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016.
[33] J.Y.-H. Ng, M.J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4694–4702, 2015.

Kenta Nishiyuki received MS degrees in Computer Sciences from Nara Institute of Science and Technology in 2010. He previously worked at Megachips Corporation, and ECC Corporation. He is currently working at Omron Corporation, and pursuing a Ph.D. degree in Computer Sciences at Chubu University. His current research interests include computer vision and machine learning.

Jia-Yau Shiau received his MS degree from Graduate Institute of Electronics Engineering, National Taiwan University in 2018. He engaged in several projects related to urban traffic management, electronic design automation, automated driving system and virtual reality, at home and abroad. He is currently working at HTC VIVE to develop VIVE tracking system. His research interests include approximation algorithm, computer vision and machine learning.

Shigenori Nagae received his Ph.D. degree in molecular biology from Graduate school of Biostudies, Kyoto University, Japan in 2013. He is currently working in OMRON Corporation. His current research interests include human activity understanding, machine learning and robotics.

Tomohiro Yabuuchi received the M.S. degree in informatics from Kyoto University, Japan in 2005. He is currently a research engineer at OMRON Corporation. Before joining OMRON Corporation in 2016, he was a specially appointed assistant professor of Osaka Institute of Technology from 2010 to 2015. Prior to that, he was JSPS research fellow (DC1) at the Academic Center for Computing and Media Studies, Kyoto University. His research interests include computer vision, machine learning, virtual reality and augmented reality. He is a member of the IEEE and the IEICE.

Koichi Kinoshita received the M.S. degree from Kobe University, Japan, in 1998 and the Ph.D. degree in informatics from Nagoya University, Japan, in 2013. He is currently working in OMRON Corporation. His current research interests include human activity understanding, machine learning and computer vision. He is a member of the IEICE.

Yuki Hasegawa received her bachelor's degree in information science from Tsukuba University, Japan in 1997. She is currently working in OMRON Corporation, as a department manager of Computer Vision. Her research interests include human understanding, 3D sensing, and robotics.

Takayoshi Yamashita received his Ph.D. degree from Department of Computer Science, Chubu University, Japan in 2011. He worked in OMRON Corporation from 2002 to 2014. He is an associate professor of Department of Computer Science, Chubu University, Japan since 2017. His research interests include object detection, object tracking, human activity understanding, pattern recognition and machine learning. He is a member of the IEEE, the IEICE and the IPSJ.

Hironobu Fujiyoshi received his Ph.D. in Electrical Engineering from Chubu University, Japan, in 1997. From 1997 to 2000 he was a post-doctoral fellow at the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA, USA, working on the DARPA Video Surveillance and Monitoring (VSAM) eort and the humanoid vision project for the HONDA Humanoid Robot. He is now a professor of the Department of Computer Science, Chubu University, Japan. From 2005 to 2006, he was a visiting researcher at Robotics Institute, Carnegie Mellon University. His research interests include computer vision, video understanding and pattern recognition. He is a member of the IEEE, the IEICE, and the IPSJ.