International Journal of Innovative Computing, Information and Control
Volume **18**, Number **5**, October **2022**

ICIC International ©2022 ISSN 1349-4198
pp. **1445–1461**

DETECTION OF DRIVER'S VISUAL DISTRACTION USING DUAL CAMERAS

Ulziibayar Sonom-Ochir[1], Stephen Karungaru[1], Kenji Terada[1] and Altangerel Ayush[2]

[1]Department of Information Science and Intelligent Systems
Faculty of Engineering
The University of Tokushima
2-1 Minami-josanjima, Tokushima 770-8506, Japan
c502147002@tokushima-u.ac.jp; { karunga; terada }@is.tokushima-u.ac.jp

[2]Department of Information Technology
Mongolian University of Science and Technology
22th Khoroo, Bayanzurkh 13340, Mongolia
a.altangerel@must.edu.mn

Received March 2022; revised July 2022

Abstract. *Most serious accidents are caused by the driver's visual distraction. Therefore, early detection of a driver's visual distraction is very important. The detection system mostly used is the dashboard camera because it is cheap and convenient. However, some studies have focused on various methods using additional equipment such as vehicle-mounted devices, wearable devices, and specific cameras that are common. However, these proposals are expensive. Therefore, the main goal of our research is to create a low-cost, non-intrusive, and lightweight driver's visual distraction detection (DVDD) system using only a simple dual dashboard camera. Currently, most research has focused only on tracking and estimating the driver's gaze. In our study, additionally, we also aim to monitor the road environment and then evaluate the driver's visual distraction detection based on the two pieces of information. The proposed system has two main modules: 1) gaze mapping and 2) moving object detection. The gaze mapping module receives video captured through a camera placed in front of the driver, and then predicts a driver's gaze direction to one of predefined 16 gaze regions. Concurrently, the moving object detection module identifies the moving objects from the front view and determines in which part of the predefined 16 gaze regions it appears. By combining and evaluating the two modules, the state of the distraction of the driver can be estimated. If the two module outputs are different gaze regions or non-neighbor gaze regions, the system considers that the driver is visually distracted and issues a warning. We conducted experiments based on our self-built real-driving DriverGazeMapping dataset. In the gaze mapping module, we compared the two methods MobileNet and OpenFace with the SVM classifier. The two methods outperformed the baseline gaze mapping module. Moreover, in the OpenFace with SVM classifier method, we investigated which features extracted by OpenFace affected the performance of the gaze mapping module. Of these, the most effective feature was the combination of a gaze angle and head position_R features. The OpenFace with SVM method using gaze angle and head position_R features achieved a 6.25% higher accuracy than the method using MobileNet. Besides, the moving object detection module using the Lukas-Kanade dense method was faster and more reliable than in the previous study in our experiments.*

Keywords: Visual distraction, Gaze mapping, Moving object, Gaze region

DOI: 10.24507/ijicic.18.05.1445

1. **Introduction.** According to the World Health Organization (WHO), over 1.3 million deaths occur worldwide each year due to traffic accidents alone making it one of the top eight causes of death. Moreover, most traffic accidents were caused by distracted driving [1]. Thus, recent studies have addressed the issue of the detection of distracted drivers. Driving is a complex process that depends on many factors and is also difficult to monitor. Therefore, it is necessary to monitor the driving process, evaluate the state of the distraction of the driver, and warn or guide the driver. This is particularly important when the driver fails to notice a pedestrian or other moving object. For example, sometimes, the driver might not notice the moving object (pedestrian, bicycle, obstacle, animal, etc.) when the vehicle is moved from a stationary state (standing at a traffic light) to a moving state shown as in Figure 1. Therefore, developing a driver's visual distraction detection (DVDD) system for evaluating a driver's visual attention is highly necessary to reduce traffic accidents caused by distracted driving. Consequently, a lot of studies have been conducted, and some of them have been implemented. These related studies can be classified according to the method of determining the driver's distraction, and three main types of data used to recognize distracted drivers. The first method was based on physiological data such as an electrocardiogram (ECG), and an electroencephalogram (EEG). The second method was based on vehicle control data such as pedal positions, and steering wheel movements. Finally, the third method was based on visual data such as eye movements, body movements, and images or videos of the driver's facial expressions [2]. Of these, the studies based on the visual data of the driver account for the majority of the work. Our study also proceeds to focus on the detection of distracted drivers using visual data. Driving is considered a complex activity that requires the driver's all sensor, cognitive, and body movement processes to ensure safety. The following are some related studies that were conducted to determine a driver's visual attention.

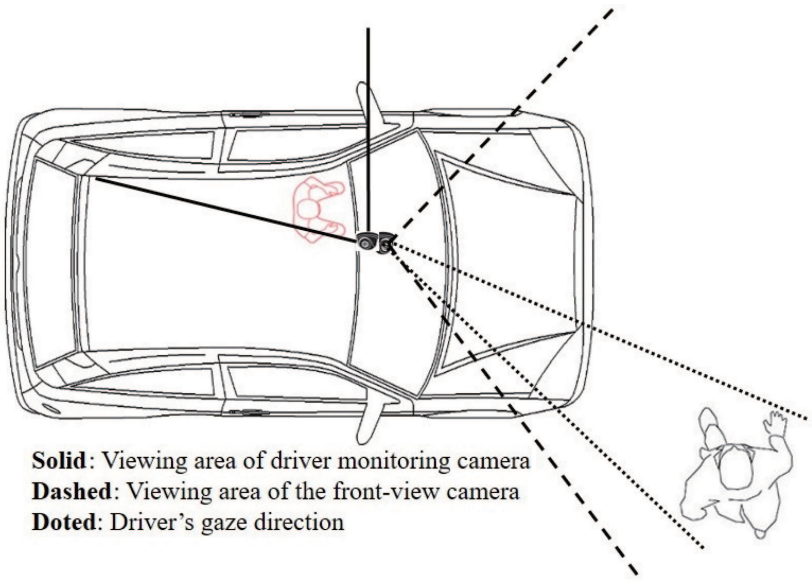


FIGURE 1. Detection of driver's visual distraction system

In recent years, driver distraction in human driving [3-8] has been carried out. This type of studies often uses a driver's behavioral indicator to determine driver status. Tsubowa et al. [8] conducted the driver status monitoring system for detecting a decrease in driver arousal by using wearable devices. The head pose and gaze estimation is the common way to estimate the driver's visual attention when driving. Several approaches based on

the front camera of the vehicle-mounted device or windshield-mounted smartphone [9,10] monitored the driver's attention by combining several features (spatio-temporal features based on the driver's state) and behavior (head pose, eye gaze, eye closure, yawns, use of cell phones). Although they provide accurate gaze information, sometimes, gaze information can be inaccurate because of the slight movement of the camera due to the use of 3D geometric reasoning. Moreover, Mizuno et al. [11] conducted a system of visual attention detection using a gaze tracker and vehicle-mounted device. Although these approaches are effective and robust, they are intrusive, costly, and unsuitable for real application. Recently, an open-source eye-tracking-based toolkit is commonly used for related studies due to the low cost and being non-intrusive because this toolkit does not require additional special equipment. Araluce et al. [12] used an open-source state-of-the-art OpenFace v2.0 toolkit [13] to monitor the driver's visual attention using gaze estimation, which is a typical approach used on the road scene. Besides, Fridman et al. [14] detected the visual features of a driver using a single camera and then estimated the driver's gaze in six different regions. Furthermore, several studies have recently been conducted to identify the visual attention of drivers in real-time conditions [15-17]. However, they used the driver monitoring camera to estimate whether the driver visual distraction, while disregarding the road situation. These studies focused on accurately determining where the driver is looking using the eye gaze direction and the driver's other features such as head pose, etc. However, it is not sufficient to detect the driver's visual distraction because driving is a complex process and depends not only on the driver but also on other road users. Determining where the driver is looking is only one module of our study and we also estimate where the moving object appears in the driver's gaze regions. If the driver does not look at the gaze region where the moving object is detected, it is considered that driver's visual distraction has occurred. In this case, the DVDD system warns the driver. Xiao and Feng [18] evaluated drivers' visual distraction based on monitoring the driver's gaze direction and the road information. It is the most similar study to our work. Therefore, we chose this study as a baseline and compared the performance of each module of our system with the corresponding module of the baseline.

Our paper proposes to evaluate the detection of driver's visual distraction based on different methods, such as MobileNet [19] and OpenFace v2.0 with a support vector machine (SVM) classifier. These methods are used to estimate a driver's gaze mapping, while the optical flow dense method is used to detect moving objects. We evaluated the state of driver distraction based on two inputs. One is a video stream of the driver's monitoring camera (for estimating gaze region), and the other input is a video stream of a front-view camera (detection of a moving object), as shown in Figure 2. We construct a detection of the driver's visual distraction system, which is low-cost, non-intrusive, and not sensitive to the slight movements of a dual camera.

Compared to related works, our contributions and achievements are as follows.

- Most studies focus on only the accuracy of gaze mapping. The main feature of our study is not only to determine the driver's visual attention (gaze) but also to monitor the road environment and then evaluate the driver's visual distraction based on the two. In addition, we proposed a low-cost, non-intrusive, and lightweight driver's visual distraction detection (DVDD) system using a simple dual dashboard camera without the need for additional equipment.
- In the first module, the gaze mapping module, we proposed and compared two methods: MobileNet and OpenFace with SVM classifier. Our methods outperformed the currently best baseline gaze mapping module. Furthermore, in the OpenFace with SVM classifier method, we investigated which features extracted by OpenFace

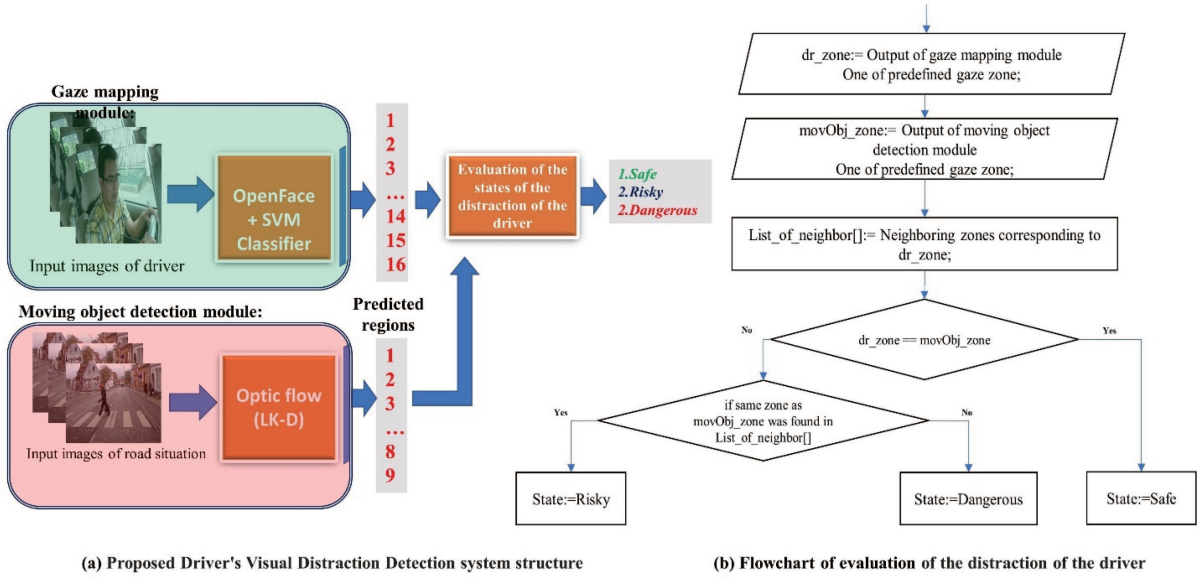


FIGURE 2. Proposed driver's visual distraction detection system structure

affected the performance of the gaze mapping module. Of these, the most effective feature was the combination of a gaze angle and head position_R features.

- In the second module, moving object detection, we compared two methods: Lucas-Kanade dense and sparse to explore which method would be effective for the moving object detection module. Experimental results demonstrated that the Lucas-Kanade dense method detected moving objects faster than the sparse method.

The rest of this paper is organized as follows. First, a description of the DVDD system, the datasets used for training, and the methods we used are provided in Section 2. Then, in Section 3, we describe our experiments and the inference result of combining the best version of the above methods for gaze mapping and the moving object detection method. Our conclusions are then presented in Section 4.

2. **System Structure and Dataset.** Our system has two modules, gaze mapping and moving object detection. The gaze mapping module receives the video captured through the driver monitoring camera, and then defines the gaze region the driver is looking at. We investigated different methods, such as using the MobileNet model and OpenFace with SVM classifier, for Gaze mapping. The moving object detection module gets a video stream of the front-view camera (outside road situations) and identifies the gaze region where the moving object is detected. We combine the two modules and evaluate the state of the distraction of the driver. We considered three states of the distraction of the driver. An algorithm for evaluating these states is shown in Figure 2(b).

a) **Safe State:** If the driver's gaze region is THE SAME as the one the moving object is detected in, it is considered a "safe" state.
b) **Risky State:** If the driver's gaze region is NOT THE SAME as the one the moving object is detected in, BUT it is a neighboring gaze region, it is considered a "risky" state.
c) **Dangerous State:** If the driver's gaze region is THE SAME as the one the moving object is detected in, and NOT a neighbor region EITHER, it is considered a "dangerous" state.

In this paper, we defined 16 gaze regions and collected data from 4 volunteer drivers. The data includes the driver's gaze and the driving environment information. The 16

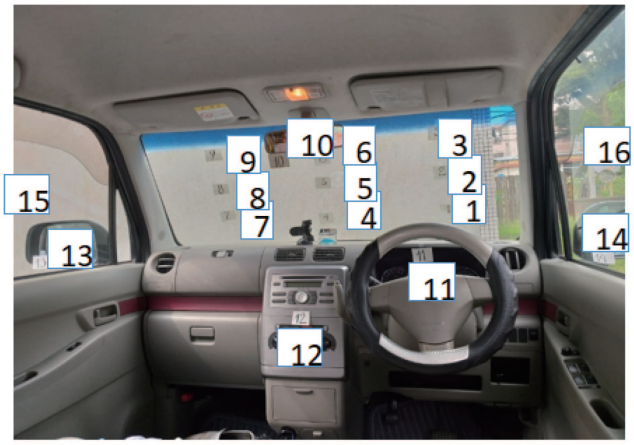


FIGURE 3. Pre-defined 16 gaze regions

TABLE 1. Gaze regions and neighbor regions of each gaze region

Gaze regions	Neighbors
1 Windshield	2, 4, 5, 11, 14, 16
2 Windshield	1, 3, 4, 5, 6, 14, 16
3 Windshield	2, 5, 6, 10, 16
4 Windshield	1, 2, 5, 7, 8, 11, 12
5 Windshield	1, 2, 3, 4, 6, 7, 8, 9, 10
6 Windshield	2, 3, 5, 8, 9, 10
7 Windshield	4, 5, 8, 13, 15
8 Windshield	4, 5, 6, 7, 9, 13, 15
9 Windshield	5, 6, 8, 10, 15
10 Rearview mirror	3, 5, 6, 9
11 Dashboard	1, 4, 12, 14
12 Music/radio	4, 7, 11, 13
13 Left side mirror	7, 8, 15
14 Right side mirror	1, 2, 11, 16
15 Left side window	7, 8, 9, 13
16 Right side window	1, 2, 3, 14

predefined gaze regions, as shown in Figure 3, are 1-9 gaze regions on the windshield, rearview mirror, left and right-side mirrors, left and right-side windows, dashboard, and music/radio gaze regions. We also considered the corresponding neighboring regions of each gaze region as shown in Table 1.

The dataset was built using images of drivers who gazed at predefined 16 regions in the vehicle. We captured the data as the vehicle went to different locations such as university campus roads and parking lots in the morning, afternoon, and night to get images at different times of the day using a simple COOAU-D30-1080P dual dash camera. As the drivers gazed at the 16 predefined regions, they naturally acted, with no restrictions on changes in the head pose or other movements. The dataset consists of 12,425 images with 16 labels, and some example samples are shown in Figure 4. We denoted this dataset as DriverGazeMapping (DGM).

2.1. **Gaze mapping module.** In the first module, we selected two methods, MobileNet and OpenFace with an SVM classifier. The main proposal of this work is to contribute

FIGURE 4. Samples of the collected data

to the distracted driver detection system based on a dash camera. We aim for a non-intrusive, cheap and lightweight system that is not difficult to use or fatigues the driver. Therefore, vehicle-mounted eye trackers and head-mounted eye trackers do not meet the requirements of our system. Consequently, face feature detection using visual data is a key task for this type of systems. OpenFace is one of the most popular open-source facial analysis tools in this research field because of its robustness and performance. It provides face detection and 68 facial features. Therefore, we use some of these features to estimate the driver's gaze in our predefined 16 gaze regions using the classifier in the gaze mapping module. As our proposed system is a real-time system, the performance speed is crucial. In terms of performance speed, the SVM classifier was much faster than other classifiers [20]. Therefore, we chose the OpenFace with an SVM classifier for the gaze mapping module.

Also, as for MobileNet, we chose it to keep our system light, because MobileNet uses less computation power to run. This makes it a perfect fit for mobile devices, light systems, and computers that run without GPUs. Also, MobileNet significantly has a lower number of parameters making it a lightweight deep neural network. It is therefore fast and well suited for our system. Then, we experimented with these methods on the gaze mapping module to compare the training data and the real-time environments. Although MobileNet did not outperform the OpenFace with the SVM classifier, it was better than the baseline gaze mapping module.

2.1.1. *Gaze mapping using the MobileNet model.* The first method for gaze mapping used the MobileNet deep learning method. We used a pre-trained MobileNetV2 model without the last dense layer. We then added a dense layer with 16 predefined gaze regions as shown in Figure 5. On the DGM dataset, the MobileNet model was trained using four different fine-tuning and transfer learning strategies. The first strategy is to train only the

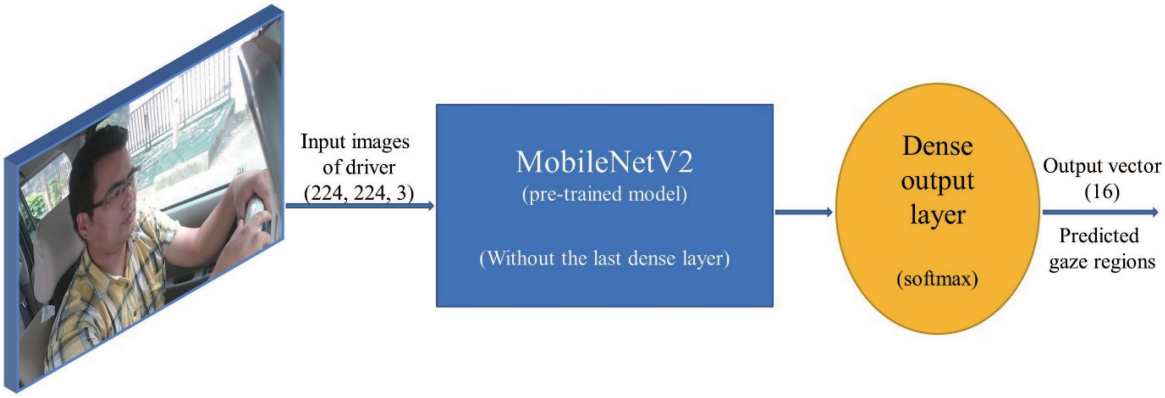


FIGURE 5. Gaze mapping module using MobileNet model

last classifier layer of the MobileNet model, while the second strategy is to train the last 30 layers of the model including the classifier layer. We trained the last 50 and 70 layers of the model including the classifier layer for the third and fourth strategies, respectively.

We noted that by increasing the number of trainable layers from 30 to 50, the accuracy of the training was improved. However, setting trainable layers to 70, the accuracy was lower. In addition, we also tested the model with 80 trainable layers, but the result was lower than others. Therefore, we do not show the result of the model with 80 trainable layers in our results. The model with 50 trainable layers achieved the best result of the experimented models.

2.1.2. *Gaze mapping using OpenFace with SVM classifier.* Determining face and head features from images is one of the challenges of gaze mapping. OpenFace is one of the robust toolkits used to extract face and head features. Therefore, we used the OpenFace toolkit, which provides gaze angles and head position features, for the gaze mapping task. The gaze angle and head position features are recognized by analyzing the driver's face from the driver monitoring camera using the OpenFace toolkit.

We also used the multi-class SVM, which is more appropriate for our task, with the OpenFace toolkit, as shown in Figure 6. We chose the SVM classifier based on the following.

- ***Performance speed:*** Our proposed system is a real-time system, so the performance speed must be high. In terms of performance speed, the SVM classifier is much faster than other classifiers [20]. Therefore, we chose the SVM classifier.
- ***The accuracy of classifying gaze mapping:*** The accuracy of classifying gaze mapping has high mean overall accuracies. This is shown in [20] and other studies

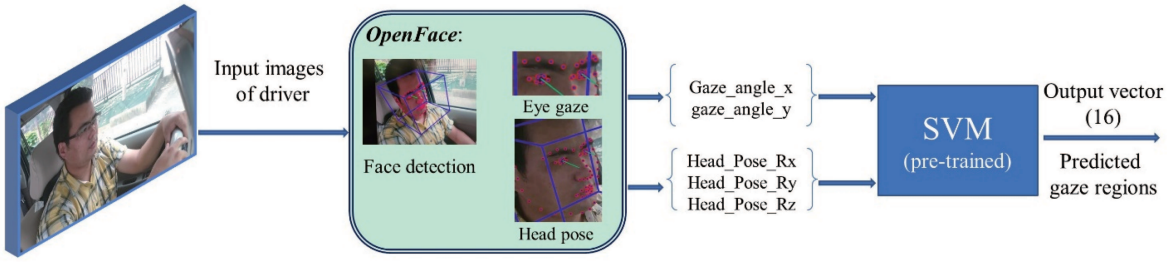


FIGURE 6. Gaze mapping module using OpenFace (using gaze angle and head position features)

[21,22]. In these studies, the SVM classifier exhibited superior results to the neural network and random forest methods in terms of overall accuracy and robustness. Therefore, we used the SVM algorithm to implement the classification of gaze mapping. In addition, SVM is better at classifying extraction data of the OpenFace, as can be seen from the study by Rill-Garcia et al. [22].

- ***Amount of our data:*** The OpenFace was used to extract gaze direction and head direction features from the DGM dataset. We then trained the SVM classifier on this dataset. Our dataset is relatively small, with few samples, making SVM classifiers more suitable.

We tuned the hyper-parameters to train the SVM classifier using GridSearchCV [23] from the Scikit Learn library. GridSearchCV helps combine an estimator with a grid search preamble to tune hyper-parameters such as kernel, C, and gamma. To determine the value of parameters C and gamma for searching for the best value, we experimented with C from 0.1 to 100 and the gamma from 0.0001 to 10. According to GridSearchCV, the most appropriate parameters for the dataset extracted from the OpenFace toolkit were defined as {C = 10, gamma: 0.1, kernel: 'rbf'}. We conducted the training of the SVM classifier in four different strategies using the features extracted by OpenFace. The features we used are as follows.

- gaze angle (gaze_angle_x, gaze_angle_y),
- head position_T (Head_Pose_Tx, Head_Pose_Ty, Head_Pose_Tz),
- head position_R (Head_Pose_Rx, Head_Pose_Ry, Head_Pose_Rz), and
- a combination of the gaze angle and head position_R.

Thereafter, we trained the SVM classifier on the four different datasets consisting of the feature and corresponding gaze region labels. A portion of the DGM dataset mentioned in Section 2 was used to generate these four datasets for the training. During the execution of the gaze mapping module, as shown in Figure 6, the gaze angle and head position features extracted by OpenFace are fed into the pre-trained SVM classifier, which then predicts one of the pre-defined 16 gaze regions corresponding to these features.

2.2. **Moving object detection.** The moving object detection is the second module of the DVDD system. In addition to the driving situations, our system also monitors outside road conditions using a front-view camera. The moving object detection module detects a moving object and then determines in which gaze region of the windshield the object appears. Sometimes, a visually distracted driver might fail to notice a pedestrian or other moving object. Moving object detection is different from object detection and tracking. Recognizing an object from video captured using a stationary camera and a moving object from a non-stationary camera environment are two different tasks. We used the optical flow method, which is more appropriate for our moving object detection task. The optical flow method can be any of two types: sparse and dense. We tested different implementations of the Optic flow method, to find out which method would be effective and meet our system's requirements as shown in Figure 7. However, methods except the Lucas-Kanade dense and sparse required too many resources that could not be used in a real-time system.

In the moving object detection module, early detection of moving objects is very important. In other words, the speed of moving object detection should be fast. The faster it is, the safer and more reliable it is. For the sparse method, the number of points required to detect a moving object is too small. In other words, the sparse method has the disadvantage of assuming that no motion is detected if the moving object does not pass through the control point. This risks the late detection of moving objects because, in traffic, unexpected events can occur suddenly. Therefore, although the sparse method

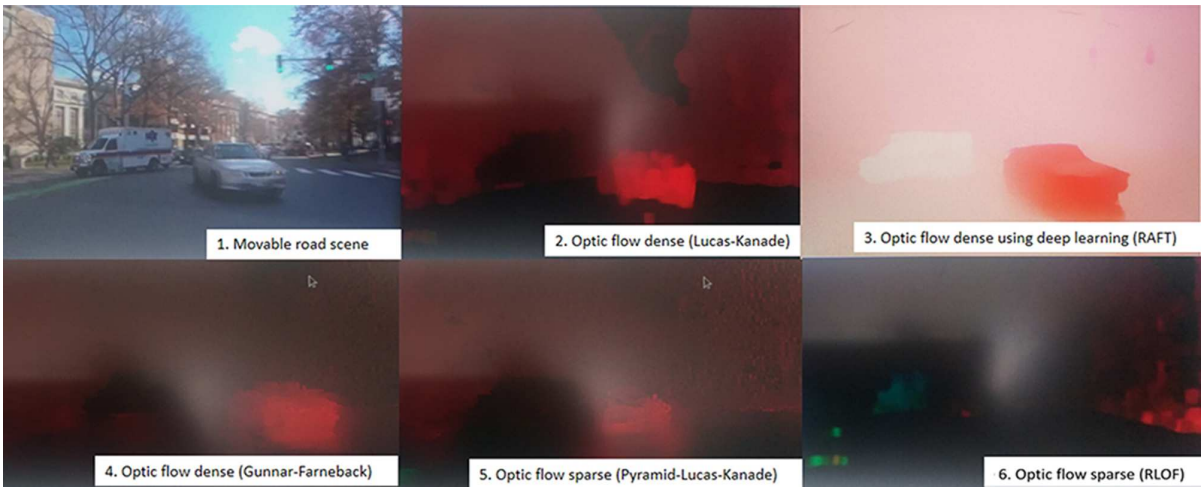


FIGURE 7. Comparative results of the optical flow methods' implementations

requires fewer resources, it does not meet the requirement of accurate and fast object detection. We chose Lucas-Kanade dense method from many implementations of Optic flow because the speed of moving object detection was faster in our experiment. We experimented with both the Lucas-Kanade dense method and sparse method (the last one was used for the moving object detection module of the baseline study) in the moving object detection module. The Lucas-Kanade dense method detected moving objects faster than the sparse method in this experiment, as shown in Figure 9 and Figure 10. The sparse method correctly detected a moving object in gaze region 2, but it was detected after passing two-thirds of the gaze region 2 (the two images in the first column of Figure 10).

In addition to the Lucas-Kanade dense and sparse methods, we tested different implementations of the Optic flow method, to find out which method would be effective and meet our system's requirements as shown in Figure 7. However, other methods required too many resources and cannot be implemented in a real-time system.

3. **Experiment and Results.** In this section, we present the evaluation of the two modules, gaze mapping and moving object detection, separately and their combination. In other words, we show the evaluation of the state of the driver's distraction based on two inputs. As we explained in Section 2, we implemented different strategies for each method of the gaze mapping module in the DVDD system. The following sections provide a more detailed comparison of the results for each method of the gaze mapping module.

3.1. **Experiment of gaze mapping and result.** First, we fine-tuned the pretrained MobileNet trained on the ImageNet dataset [24] with our DGM dataset. Table 2 shows the results of the MobileNet models using different strategies. We evaluated the performance of each model using 180 drivers' face images for each gaze region. The model with 50 trainable layers in Table 2 achieved the best result at 97.45% accuracy.

TABLE 2. Results of the MobileNet models using different strategies

Strategies	Accuracy	Precisions	Recall	F1 score
Only classification layer	92.64%	92.97%	92.64%	92.60%
Last 30 layers	93.56%	94.16%	93.56%	93.56%
Last 50 layers	**97.45%**	**97.46%**	**97.45%**	**97.45%**
Last 70 layers	96.86%	96.91%	96.89%	96.90%

Secondly, we used the OpenFace toolkit to estimate gaze angle and head position features, and then classified the driving gaze direction into 16 pre-defined gaze regions using the SVM classifier. The results of the SVM classifier using different features mentioned in Section 2.1.2 are shown in Table 3. We evaluated the performance of each strategy using the same data used in the evaluation of the MobileNet model.

Table 3. Results of the SVM classifier using different features

Strategies / Gaze regions	OpenFace + SVM using gaze angle	OpenFace + SVM using head pose_T	OpenFace + SVM using head pose_R	OpenFace + SVM using gaze angle and head pose_R
	Accuracy			
1	71%	22%	76%	78%
2	24%	53%	69%	68%
3	83%	37%	80%	80%
4	55%	25%	45%	61%
5	36%	2%	55%	47%
6	74%	46%	67%	85%
7	80%	25%	71%	80%
8	63%	0%	63%	68%
9	74%	40%	71%	74%
10	38%	91%	77%	81%
11	22%	0%	10%	20%
12	32%	62%	74%	71%
13	86%	30%	91%	91%
14	12%	57%	86%	88%
15	91%	9%	91%	91%
16	25%	0%	99%	99%
Overall	54.12%	31.18%	70.31%	**73.87%**

When using the gaze angle feature (OpenFace + SVM using gaze angle), the accuracy of the SVM classifier for gaze regions 2, 11, 14, and 16 was lower than others, and the average accuracy was 54.12%. The OpenFace extracts two different head positions, head position_T, and head position_R. In the case of using the head position_T feature (OpenFace + SVM using head pose_T), the accuracy of the SVM classifier for gaze regions 8, 11, and 16 was not recognized, and the average accuracy was 31.18%. The performance of the SVM classifier using the head position_R (OpenFace + SVM using head pose_R) was higher than the previous two strategies. When using a combination of gaze angle and head position_R features (OpenFace + SVM using gaze angle and head pose_R), the SVM classifiers outperformed the other strategies. The average accuracy of this strategy was 3.56% higher than the previous best-performing SVM classifier using the head position-R feature.

In the OpenFace with SVM classifier experiments, we tested different features, such as gaze angle, head position_T, and head position_R. Experimental results showed that the gaze angle and head position_R features were more effective than the head position_T feature. In addition, the combination of gaze angle and head position_R features was more effective than when using each feature separately.

Finally, we chose the best strategy for each method and evaluated the performance of each strategy using a real driving video. Table 4 shows the results of these evaluations.

The strategy of the gaze mapping using the MobileNet model in Table 4 predicted all of the gaze regions except 11 and 12 with high accuracy. The strategy of the gaze mapping using the OpenFace with SVM classifier (using gaze angle and head position_R features) predicted all gaze regions. The MobileNet model performed better than the OpenFace with SVM classifier when using test data (180 drivers' face images for each gaze region). However, when using the real driving video, the average accuracy of the strategy using the OpenFace with SVM classifier was 6.25% higher than the strategy using MobileNet. Therefore, we chose the OpenFace with SVM classifier to compare with the existing system, which is our baseline.

Table 4. Results of the best strategy for each method

Methods / Gaze regions	MobileNet /Last 50 layers/	OpenFace + SVM /gaze angle + head pose_R/
1	3/3	2/3
2	3/3	2/3
3	3/3	3/3
4	2/3	3/3
5	2/3	3/3
6	3/3	3/3
7	2/3	3/3
8	2/3	3/3
9	3/3	1/3
10	3/3	3/3
11	0/3	3/3
12	0/3	3/3
13	3/3	3/3
14	3/3	3/3
15	3/3	1/3
16	3/3	2/3
Overall (%)	79.16	**85.41**

3.2. **Results of the comparison of the gaze mapping module with the existing system.** In this section, we compared our best strategies for the gaze mapping module with the front camera module of the baseline study [18]. Note that the camera position is above the driver's control panel in [18], while it is behind the middle mirror in our study. The camera position is very essential for the system to work, as it allows monitoring of a wider range of driving environments. Araluce et al. [12] considered that the best accuracy was obtained when the camera was just in front of the user at 48 cm above the table level. It is closest to the middle mirror position. The camera position we chose based on [12] has the advantage of not non-interference with the driving.

Table 5 shows the results of comparing our best strategy with the front camera module of the baseline trained on the DGM dataset. Our method correctly identified most of the gaze regions seen by the driver, although some gaze regions were confused with neighboring states.

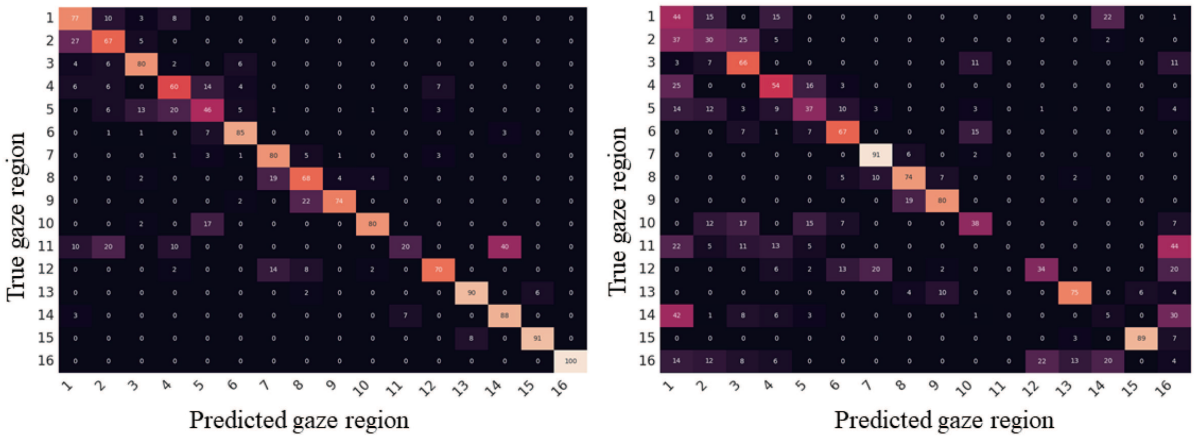
TABLE 5. Results of comparison of the gaze mapping module (OpenFace with SVM using gaze angle + head pose_R features) of our system with the front camera module (baseline) trained on the DGM dataset

Systems / Gaze regions	Gaze mapping module (OpenFace with SVM using gaze angle + head pose_R features)	The front camera module (baseline)
	Accuracy	
1	78%	43%
2	68%	30%
3	80%	66%
4	61%	53%
5	47%	37%
6	85%	66%
7	80%	91%
8	68%	74%
9	74%	80%
10	81%	38%
11	20%	0%
12	71%	34%
13	91%	75%
14	88%	5%
15	91%	89%
16	99%	4%
Overall	**73.87%**	49.06%

On the contrary, experimental results show that the front camera module based on the eye's pupil could not identify some gaze regions. In other words, it is less effective in situations where no face is detected. Notably, in gaze regions 11, 14, and 16, the results are lower than in the others because the driver's face is almost invisible to the camera when the driver looks at these gaze regions. The confusion matrix in Figure 8 shows details. Therefore, it indicates that our method, the OpenFace with SVM classifier, using gaze angle and head position_R features is more effective.

3.3. **Combination of gaze mapping and moving object detection modules.** In this section, we evaluated the combination of the best version of each gaze mapping method and the Lucas-Kanade dense method, which was used for moving object detection. In addition, the baseline system was also evaluated to compare the performance with our system. We evaluated the state of the distraction of the driver based on the combination of two modules, such as gaze mapping and moving object detection, on the inference video. In this inference video, the moving object moves along gaze regions 2, 5 and 8, while the driver looks at the object. Figure 9 shows the evaluation scene of the combination of gaze mapping and moving object detection modules. Besides, the detection of each module in each system is shown in Table 6.

The MobileNet model determined the direction of a driver's gaze, gaze regions 2 and 5 were correctly, but gaze region 8 was incorrectly predicted as gaze region 10. On the other hand, when the gaze mapping method using the OpenFace with SVM classifier determined the direction of a driver's gaze, gaze regions 8 and 5 were correctly predicted, but gaze region 2 was incorrectly predicted as gaze region 14. We observed that the incorrectly predicted gaze region was a non-neighboring of the target gaze region in the first method.



(a) (b)

FIGURE 8. Confusion matrixes of (a) the gaze mapping module (OpenFace with SVM using gaze angle + head pose_R features) and (b) the front camera module (baseline)

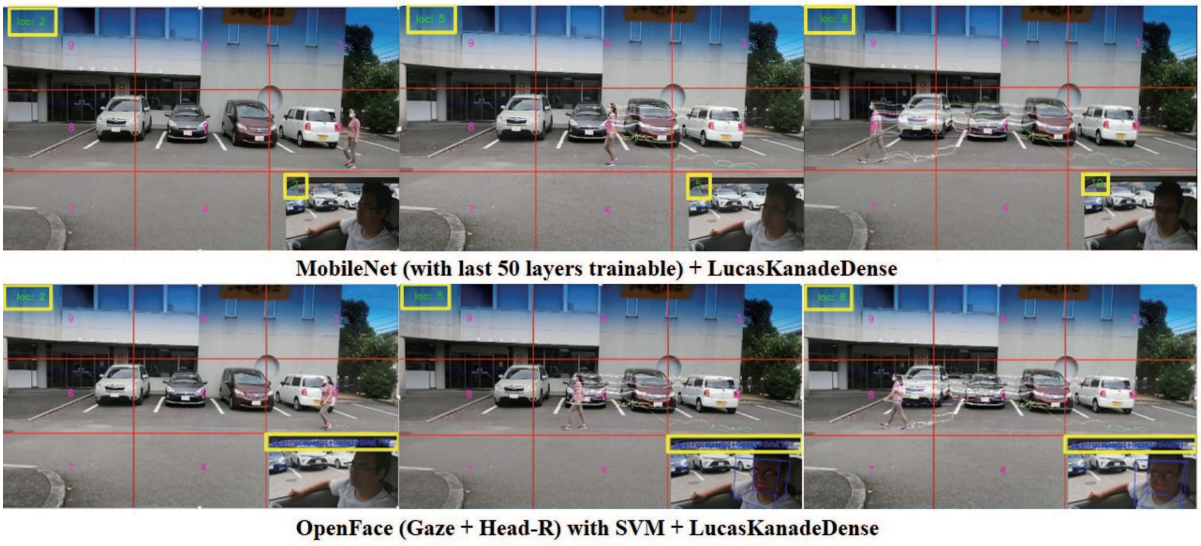


FIGURE 9. Evaluation scene of a combination of gaze mapping and moving object detection modules

In contrast, the incorrectly predicted gaze region was a neighbor of the target gaze region in the second method. In general, the performance of the gaze mapping method using OpenFace with SVM classifier (using gaze angle and head position_R features) achieved the best performance in all experiments we conducted. Besides, the Lucas-Kanade dense method for the moving object detects the module correctly and predicted all of the gaze regions where the moving object appeared.

We also evaluated a combination of the front camera module (gaze mapping) and the rear camera module (moving object detection) of the baseline using the same inference video. The gaze mapping module of the baseline determined the direction of a driver's gaze, gaze regions 2 and 5 incorrectly, and gaze region 8 was incorrectly predicted as neighboring gaze region 15, as shown in Figure 10.

Meanwhile, the moving object detection modules of the baseline correctly recognized gaze regions 5 and 8 where the moving object appeared. Even if a moving object was

TABLE 6. Comparative results of a combination of gaze mapping and moving object detection (MOD) modules of our systems and baseline

Systems	Evaluated gaze regions								
	2	5	8	2	5	8	2	5	8
	Gaze mapping			Moving object detection			States		
Baseline	No	No	Neighbor	YesNo	Yes	Yes	Dangerous	Dangerous	Risky
Our System 1	Yes	Yes	No	Yes	Yes	Yes	Safe	Safe	Dangerous
Our System 2	Neighbor	Yes	Yes	Yes	Yes	Yes	Risky	Safe	Safe

Baseline = Front camera module + rear camera module (Lucas Kanade sparse)
Our System 1 = MobileNet (with last 50 layers trainable) + MOD (Lucas Kanade dense)
Our System 2 = OpenFace with SVM (gaze + head_R) + MOD (Lucas Kanade dense)
Yes = correctly recognized/predicted gaze region
No = incorrectly recognized/predicted gaze region
Neighbor = incorrectly recognized/predicted gaze region, but neighboring gaze region

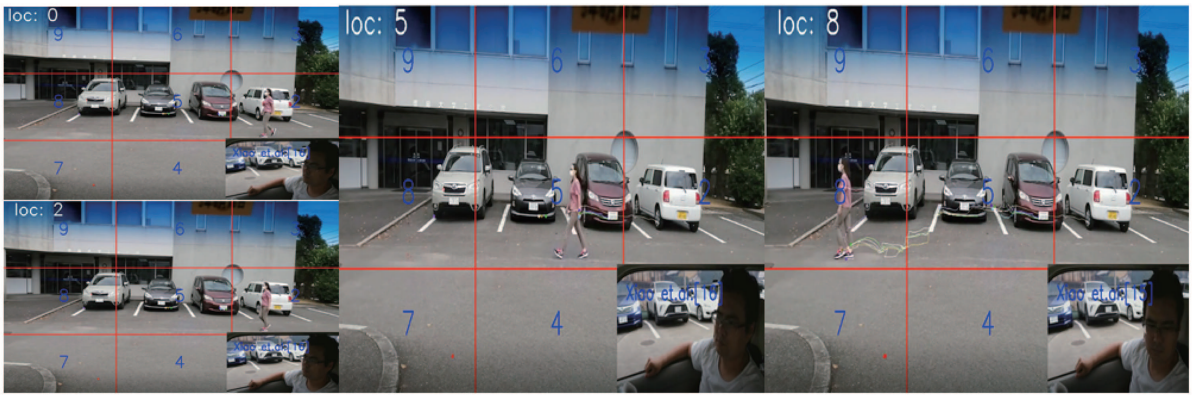


FIGURE 10. A baseline study's evaluation scene of a combination of gaze mapping and moving object detection modules

correctly detected in gaze region 2, it was detected after passing two-thirds of the gaze region 2 (the two images in the first column of Figure 10). Therefore, we marked the result of the moving object detection module for gaze region 2 as YesNo. In addition, we chose Lucas-Kanade dense method for the moving object detection module, while the baseline used the Lucas-Kanade sparse method. The Lucas-Kanade dense method detected moving objects faster than the sparse method in this experiment, as shown in Figures 9 and 10.

4. **Conclusions.** The main proposal of this work is to contribute to the distracted driver detection system based on a dash camera. It is a non-intrusive, cheap, and lightweight system that is easy to use and does not cause weariness in the driver. Our system consists of two modules, gaze mapping and moving object detection module. We proposed different methods for the gaze mapping module. The method using OpenFace with SVM classifier (using gaze angle and head position_R features) outperformed all other methods. The result was also 6.25% higher than the other method using MobileNet when using a real driving video. This is because the MobileNet model was sensitive to small changes in the camera position. Moreover, our best performing strategies of both methods (OpenFace

with SVM classifier and MobileNet) for gaze mapping outperformed the gaze mapping module of the baseline. The gaze mapping module of the baseline is less effective in situations where no face is detected. On the contrary, the gaze mapping module of our system can determine the gaze direction of the driver using a head feature, even when the face is not detected. Besides, in the moving object detection module, experimental results demonstrate that the Lucas-Kanade dense method detected moving objects faster than the sparse method. Experiments proved that our methods are effective.

In future work, we will prepare more driver's monitoring and driving situation paired data and investigate other approaches for gaze mapping and moving object detection to improve the performance of our current system.

REFERENCES

[1] World Health Organization, *Road Traffic Injuries*, http://www.who.int/news-room/fact-sheets/detail/road-traffic-injures, Accessed on August 23, 2021.

[2] M. H. Alkinani, W. Z. Khan and Q. Arshad, Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements, and open challenges, *IEEE Access*, vol.8, pp.105008-105030, 2020.

[3] C. Huang, X. Wang, J. Cao, S. Wang and Y. Zhang, HCF: A hybrid CNN framework for behavior detection of distracted drivers, *IEEE Access*, vol.8, pp.109335-109349, 2020.

[4] A. Behera, Z. Wharton, A. Keidel and B. Debnath, Deep CNN, body pose, and body-object interaction features for drivers' activity monitoring, *IEEE Trans. Intelligent Transportation Systems*, pp.1-8, 2020.

[5] M. Martin, M. Voit and R. Stiefelhagen, An evaluation of different methods for 3D-driver-body-pose estimation, *Proc. of the IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, United States, pp.1578-1584, 2021.

[6] H. M. Eraqi, Y. Abouelnega, M. H. Saad and M. N. Moustafa, Driver distraction identification with an ensemble of convolutional neural networks, *J. Adv. Transp.*, vol.2019, pp.1-12, 2019.

[7] M. Alotaibi and B. Alotaibi, Distracted driver classification using deep learning, *J. Signal, Image, and Video Processing*, vol.14, no.3, pp.617-624, 2019.

[8] K. Tsubowa, T. Akiduki, Z. Zhang, H. Takahashi and Y. Omae, A study of effects of driver's sleepiness on driver's subsidiary behaviors, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1791-1799, 2021.

[9] I. Dua, A. U. Nambi, C. V. Jawahar and V. N. Padmanabhan, Evaluation and visualization of driver inattention rating from facial features, *IEEE Trans. Biometrics, Behavior, and Identity Science*, vol.2, no.2, pp.98-108, 2019.

[10] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang and D. Levi, Driver gaze tracking and eyes off the road detection system, *IEEE Trans. Intelligent Transportation Systems*, vol.16, no.4, pp.2014-2027, 2015.

[11] N. Mizuno, A. Yoshizawa, A. Hayashi and T. Ishikawa, Detecting driver's visual attention area by using the vehicle-mounted device, *Proc. of the IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Oxford, United Kingdom, pp.346-352, 2017.

[12] J. Araluce, L. M. Bergasa, M. Ocana, E. Lopez-Guillen, P. A. Revenga and O. Perez, Gaze focalization system for driving applications using OpenFace 2.0 toolkit with NARMAX algorithm in accidental scenarios, *J. Sensors*, vol.21, no.18, pp.1-19, 2021.

[13] T. Baltrusaitis, A. Zadeh, Y. C. Lim and L. P. Morency, OpenFace 2.0: Facial behavior analysis toolkit, *Proc. of the IEEE International Conference on Automatic Face & Gesture Recognition*, Xi'an, China, pp.59-66, 2018.

[14] L. Fridman, P. Langhans, J. Lee and B. Reimer, Driver gaze region estimation without use of eye movement, *IEEE Intelligent Systems*, vol.31, no.3, pp.49-56, 2016.

[15] M. Leekha, M. Goswami, R. R. Shah, Y. Yin and R. Zimmermann, Are you paying attention? Detecting distracted driving in real-time, *Proc. of the IEEE 5th International Conference on Multimedia Big Data*, Singapore, pp.171-180, 2019.

[16] B. Baheti, S. Talbar and S. Gajre, Towards computationally efficient and realtime distracted driver detection with MobileVGG network, *IEEE Trans. Intelligent Vehicles*, vol.5, no.4, pp.565-574, 2020.

[17] D. Tran, H. M. Do, J. Lu and W. Sheng, Real-time detection of distracted driving using dual cameras, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, pp.2014-2019, 2020.
[18] D. Xiao and C. Feng, Detection of drivers' visual attention using smartphones, *Proc. of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, Changsha, China, pp.630-635, 2016.
[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp.4510-4520, 2018.
[20] I. Nitze, U. Schulthess and H. Asche, Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to the maximum likelihood for supervised crop type classification, *Proc. of the 4th GEOBIA*, Rio De Janeiro, Brazil, pp.35-45, 2018.
[21] M. Monaro, S. Maldera, C. Scarpazza, G. Sartori and N. Navarinb, Detecting deception through facial expressions in a dataset of videotaped interviews: A comparison between human judges and machine learning models, *Computers in Human Behavior*, vol.127, DOI: 10.1016/j.chb.2021.107063, pp.1-10, 2021.
[22] R. Rill-Garcia, H. Escalante, L. Villasenor-Pineda and V. Reyes-Meza, High-level features for multimodal deception detection in video, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, 2019.
[23] F. Pedregosa, G. Varoquaux, A. Gramfort and V. Michel, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research*, vol.12, pp.2825-2830, 2011.
[24] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and F.-F. Li, ImageNet: A large-scale hierarchical image database, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, pp.248-255, 2009.

Author Biography

Ulziibayar Sonom-Ochir is studying for a doctorate at the Department of Information Science and Intelligent Systems, Tokushima University. His research interests include image processing, pattern recognition, real-time image/video processing, object detection, and computer vision.

Stephen Karungaru received a Ph.D. in information system design from the Department of Information Science and Intelligent Systems, The University of Tokushima in 2004.

Dr. Karungaru is currently an associate professor in the same department. His research interests include pattern recognition, neural networks, evolutionary computation, image processing, and robotics.

Kenji Terada received a doctorate from Keio University in 1995. In 2009, he became a Professor in the Department of Information Science and Intelligent Systems, University of Tokushima department. His research interests are in computer vision and image processing. He is a member of the IEEE, IEEJ, and IEICE.

Altangerel Ayush received a doctorate in computer science from the Mongolian University of Science and Technology, Mongolia in 2011. In 2014, he became a Professor in the Department of Information Technology, School of Information and Communication Technology. His research interests are in neural networks, machine learning and image processing.