IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 May 2022, accepted 25 June 2022, date of publication 29 June 2022, date of current version 11 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187185

RESEARCH ARTICLE

Driver Behaviors Recognizer Based on Light-Weight Convolutional Neural Network Architecture and Attention Mechanism

DUY-LINH NGUYEN, (Member, IEEE), MUHAMAD DWISNANTO PUTRO, (Member, IEEE), AND KANG-HYUN JO, (Senior Member, IEEE)

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea

Corresponding author: Kang-Hyun Jo (acejo@ulsan.ac.kr)

This work was supported by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003.

ABSTRACT Driving is a set of behaviors that need high concentration. Sometimes these behaviors are dominated by other acts such as smoking, eating, drinking, talking, phone calls, adjusting the radio, or drowsiness. These are also the main causes of current traffic accidents. Therefore, developing applications to warn drivers in advance is essential. This research introduces a light-weight convolutional neural network architecture to recognize driver behaviors, helping the warning system to provide accurate information and to minimize traffic collisions. This network is a combination of feature extraction and classifier modules. The feature extraction module uses the advantages of the standard convolution layers, depthwise separable convolution layers, average pooling layers, and proposed adaptive connections to extract the feature maps. The benefit of the convolution block attention module is deployed in the feature extraction module that guides the network in learning the salient features. The classifier module is comprised of a global average pooling and softmax layer to calculate the probability of each class. The overall design optimizes the network parameters and maintains classification accuracy. The entire network is trained and evaluated on three benchmark datasets: the State Farm Distracted Driver Detection, the American University in Cairo version 1, and the American University in Cairo version 2. As a result, the accuracies on overall classes (ten classes) are 99.95%, 95.57%, and 99.61%, respectively. Also, several video tests with VGA (Video Graphics Array), HD (High Definition), and FHD (Full High Definition) resolution were conducted, and they can be seen at https://bit.ly/3GY2iJl.

INDEX TERMS Attention mechanism, convolutional neural network, driver behavior recognizer, driver warning system.

I. INTRODUCTION

Nowadays, road traffic systems have grown much in terms of quantity and complexity. Accordingly, the number of accidents also increased gradually. The statistics of the World Health Organization point out that about 1.35 billion people die and approximately 50 million road traffic collisions occur every year [1]. One of the common causes that leads to an increase in accidents is driver behavior. The statement above also mentioned that if the drivers really focused when driving, it could reduce the accident rate by four times. According to a statistic from the National Highway Transportation and Safety Administration (NHTSA) in the United States (US), about 2,895 people were killed in distracted driving accidents in 2019, accounting for 8.7% of all traffic accident deaths in that year [2]. These reports show that from 2010 to 2019, the number of deaths and accidents caused by distracted driving still maintained a quite high rate between 8% and 10%, 14%

The associate editor coordinating the review of this manuscript and approving it for publication was R. K. Tripathy.

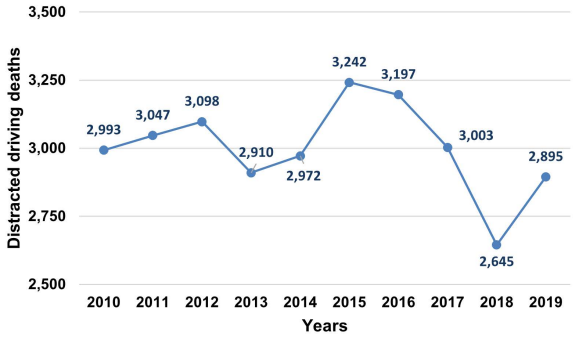

FIGURE 1. The statistics of distracted driving deaths in US within ten years.

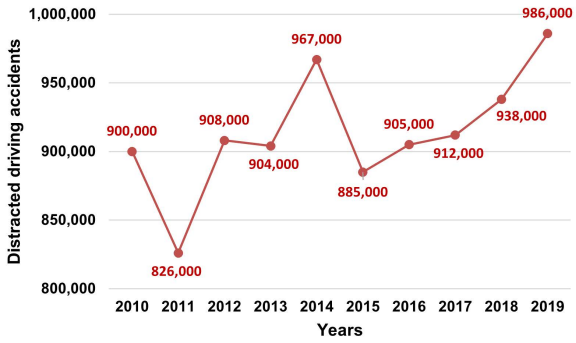

FIGURE 2. The statistics of distracted driving accidents in US within ten years.

and 16% of all accidents, respectively. The detailed number of distracted driving deaths and accidents in ten years is presented in Figure 1 and Figure 2. The scientists focused on researching the problems, solutions of road traffic accidents, and giving some definitions of distracted driving. The authors in [2] define distracted driving as driver behaviors that interrupt the focus from driving including operating cellphones (talking, texting), eating, drinking, and adjusting the entertainment system (radio, stereo). From another definition in [3], anything that distracts from paying attention to driving can be considered distracted driving. This work also divides distracted driving into three groups: visual, manual, and cognitive distraction. Visual distraction focuses on eye status and head posture analysis. The main devices for measuring visual distraction are different sensors/cameras mounted on vehicles or directly attached to the driver's body to collect signals, then analyze and process. In manual distraction, the devices are designed to track the operation of the driver's hand or foot activities on the gas and brake pedal. Cognitive distraction predicts the psychophysiological state of drivers like heart rate, blood pressure, and body temperature. Based on the above observations, automakers have integrated analytical and driver warning devices in modern cars [4]–[6]. However, most are still in the process of testing. On the other hand, most of these devices have high costs and are difficult to deploy in old cars. In addition, wearable devices are disadvantaged by safe driving operations, and the obtained signals can be affected by a few natural structures of the human body. With the goal of simplifying the devices, making them non-invasive for drivers, and saving costs, this work proposes a solution to recognize driver behaviors through a simple convolutional neural network (CNN) architecture combined with the attention technique. The proposed network takes advantage of the standard convolution layers, depthwise separable convolution layers, average pooling layers, and adaptive connections with the convolution block attention module (CBAM) to extract the feature maps then learn the outstanding features through the attention mechanism. Finally, the classifier module applies the global average pooling (GAP) layer and softmax function to compute ten probabilities of corresponding driver behaviors in the datasets. The core contributions of this research are as follows:

1) A light-weight convolutional neural network for driver behavior recognition was proposed which supports the driver warning system. This network consists of feature extraction and classifier modules. The design applies basic components in a CNN, with the proposed adaptive connections, and a convolution block attention module to learn important information of feature maps. Besides, it uses a global average pooling to replace all fully connected layers in common classification networks. Therefore, it optimizes the network parameters while maintaining high speed and accuracy. This design is suitable for implementation on low-cost and low-computation equipment, including deployment on older vehicles, without any additional installation and redesign costs. On the other hand, due to the use and process of the image signals from the camera, it is not invasive to the driver's psychology.

2) The proposed network was comprehensively trained, evaluated, and reported on all three benchmark datasets. In addition, this work built the application for video testing on different devices: GPU, CPU, and Jetson Nano.

II. RELATED WORK

This section will present several techniques applied to driver behavior recognition and their advantages and disadvantages. These methods are considered based on two respects: traditional machine learning and CNN-based methodology.

A. TRADITIONAL MACHINE LEARNING METHODOLOGY

The first research focused on detecting cellphone usage during driving. Ref. [7] uses the Supervised Descent method, a Histogram of Oriented Gradients (HOG), and an Adaboost classifier to realize the actions of using a cellphone with the accuracy of 93.9%. This study is limited by the cell phone region extraction from facial landmark technique, illumination, and occlusion conditions. Other studies measure the relative distance between four components such as the face, mouth, hands, and cellphone using Hidden CRF [8] and Support Vector Machines (SVM) [9] to classify cellphone

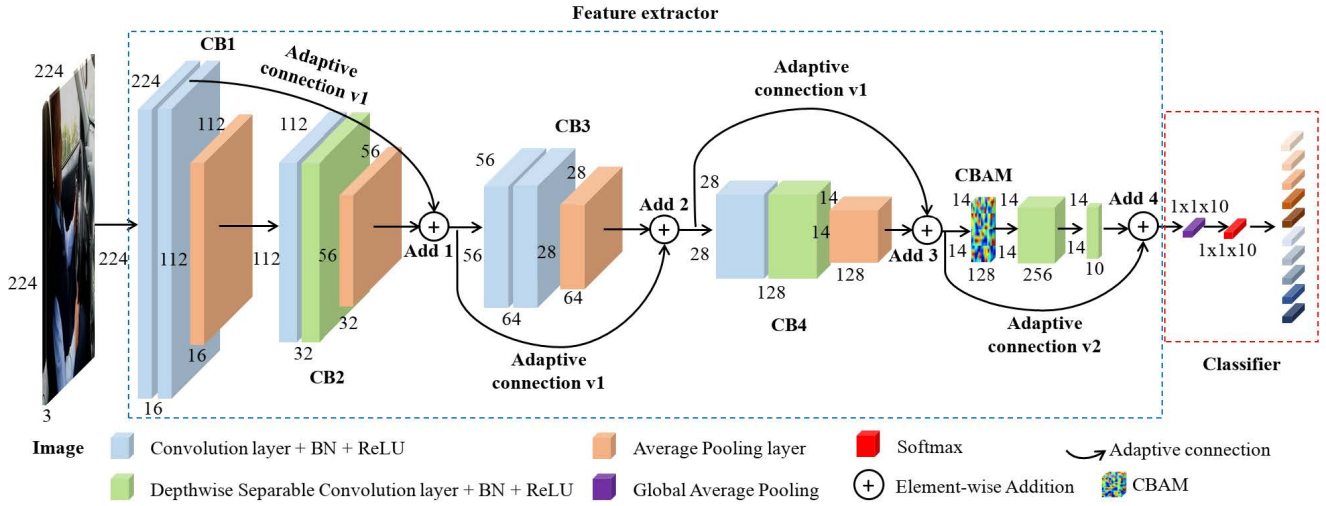

FIGURE 3. The proposed driver behaviors classification network. It consists of two main modules: the feature extractor and the classifier.

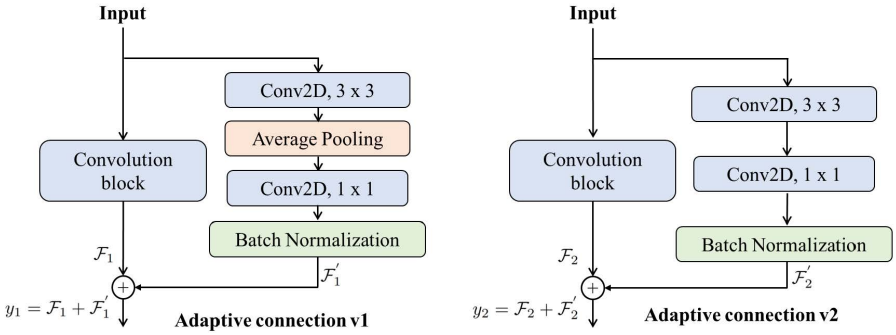

FIGURE 4. The architecture of the adaptive connections. Adaptive connection v1 (left side) with an average pooling layer between two convolution layers and adaptive connection v2 (right side) without an average pooling layer.

use. These approaches can achieve the accuracy of 91.2% and 91.57% respectively but are still mainly dependent on the lighting condition of the skin. The work in [10] also uses the SVM technique to recognize the use of cellphones in images collected from cameras mounted on the highway and traffic lights and get the accuracy about 86.19%. This study is just implemented on a small dataset with 1,500 images and the accuracy is quite low. The authors in [11] proposed a method using Hidden Markov models and an Adaboost classifier to classify images that simulate the cellphone use obtained from the RGB-D sensors of Kinect devices. An accuracy for distracted driving is 90% however the system is built based on many complex modules. In different approaches, [12] focuses on detecting the movement of the driver's hand using the Aggregate Channel Features (ACF) method with best prediction result is 70.09% of AP (Average Precision), and classifying multi-actions using the Contourlet Transform combined with Random Forests classification [13] by 88% of the highest accuracy in eating action. In summary, all traditional machine learning methods are easy to deploy but have low classification accuracy.

B. CNN-BASED METHODOLOGY

In recent years, convolutional neural networks have been widely applied in computer vision fields. Studies on human behavior in general, and driver behavior in particular, have also taken advantage of convolutional neural networks to build monitoring and warning applications. The application areas range from image detection and image segmentation to image classification. The work in [14] uses a Faster R-CNN network as a detector to guess the hand movements on the steering wheel. The results show that this method achieves an accuracy of 92.4% and 91% respectively for cell phone usage and hands on the steering wheel cases. Ref. [15] applies the image segmentation method to localize the steering wheel, gear lever, and dashboard. After that, they propose a network architecture to detect the driver's hand position on previously segmented regions and achieve 74.3% of accuracy. This combined method can solve the problem of illumination

changes but is computationally complex. In the image classification task, [16] first proposed a dataset for distracted driving classification called the Southeast dataset with four classes: smoking or eating, talking on the phone, safe driving, and operating the gear lever. This dataset is used in the traditional machine learning methods, and [17] applied several techniques with convolutional neural networks to classify these four classes with an overall accuracy of 99.78%. Later, [18]–[20] proposed extended datasets for driving distraction with ten classes (the detailed descriptions are shown in the dataset subsection). Based on these datasets, many studies have used different CNN network architectures for training and evaluation. Also, in [19] and [20], the authors propose an ensemble training method with five different CNN networks and result in an accuracy of 94.29% and 93.65%, respectively. Other typical classification neural networks such as VGG [21], [22], DenseNet [23], GoogleNet [24] have also been exploited for driver behaviors classification with the accuracy form 95% to over 99%. For the purpose of reducing network parameters and deploying low-computation devices, [25], [26] proposed convolutional neural network architectures with depthwise separable convolution operation and a residual network to classify ten driver behaviors. These methods achieve very high accuracy (over 95%) and the small number of network parameters (less than 0.5M parameters). The above studies have high accuracy, but only focused on recognizing driver behavior with a limited set of classes (four classes) or only evaluated on individual datasets with a larger number of classes (ten classes). On the other hand, several proposed methods are heavyweight and difficult to apply in real-time systems. As a study on the strength of standard convolutional and depthwise separable convolution layers, and Inception and Residual networks, this work proposes a light-weight driver behavior classification convolutional neural network. It has just 0.43M parameters but the network guarantees high accuracy when compared with many other methods.

III. PROPOSED METHODOLOGY

The proposed network architecture consists of the feature extractor and classifier modules. The feature extractor module is designed based on the stem module, adaptive connections (AC), and a CBAM to extract the feature maps. At the final feature map, the classifier module applies a GAP and a softmax function to calculate the probability of ten driver behaviors and then classifies them. Figure 3 describes in detail the proposed architecture.

A. FEATURE EXTRACTOR MODULE

Most of the popular convolutional neural networks can extract high-level feature maps from raw pixels without any manual processing steps. Therefore, later tasks such as image classification, object detection, and image segmentation will be easily applied and achieve high precision. Meanwhile, traditional machine learning methods rely heavily on image preprocessing and feature extraction, so the received precision is unstable. This study focuses to design the feature extractor based on many novel techniques to obtain the most effective feature maps. The feature extractor includes four convolutional blocks (CBs), four ACs, one CBAM, and two depthwise separable convolution layers. The CBs have two different architectures. The first architecture is built based on two standard convolution layers, a batch normalization (BN) layer, a rectified linear unit (ReLU) activation function, and an average pooling layer (in CB1, CB3). The other architecture uses a standard convolution layer, a depthwise separable convolution layer, a BN layer, a ReLU activation function, and an average pooling layer (in CB2, CB4). The kernel sizes and number of channels of convolutional blocks vary from $7 \times 7 \times 16$ (CB1), $5 \times 5 \times 32$ (CB2), $3 \times 3 \times 64$ (CB3), and $3 \times 3 \times 128$ (CB4). Applying the big kernel sizes at the beginning of the network to enlarge the receptive fields, helps the feature extractor to accurately capture the basic object information in the image. However, this work also increases the computational cost of the network. That is also the reason to use depthwise separable convolution layers later to balance the previous computation cost. After going through four convolution blocks, the $224 \times 224 \times 3$ input image will be reduced by 16 times, generating a $14 \times 14 \times 128$ feature map. This process loses a lot of important information. Therefore it is necessary to combine the information between the current feature map level and the previous feature map levels. This maintains and enriches the necessary information for all feature map levels. Inspired by ResNet [27] and Inception [28] networks, this work proposes adaptive connections with two different approaches as depicted in Figure 4. Adaptive connection version 1 (ACv1) includes a 3×3 standard convolution layer, an average pooling layer, and a 1×1 standard convolution layer followed by a BN layer. Adaptive connection version 2 (ACv2) is almost the same as ACv1 but it does not use the average pooling layer in between the two standard convolution layers. The adaptive connections serve as a branch that extracts sub-features from the previous feature map level and then combines them with the current feature map through addition. The proposed network uses four adaptive connections at different levels. Specifically, the first ACv1 is applied at the output of the second convolution layer in CB1 and adds the output of CB2. The second one (ACv2) is applied at the output of the previous addition (Add 1) and adds the output of CB3. A similar process for the third adaptive connection (ACv1) and the fourth adaptive connection (ACv2) generate the output of Add 3 and Add 4. The equations of these connections are shown as follows:

$$y_i = \mathcal{F}_i(x) + \mathcal{F}'_i(x), \tag{1}$$

where x and y are the input and output feature maps, respectively. i is the adaptive connection version ($i = 1$ or $i = 2$). $\mathcal{F}_i(x) \in \mathbb{R}^{W \times H \times C}$ is the output feature map of each convolutional block. $\mathcal{F}'_i(x) \in \mathbb{R}^{W \times H \times C}$ is the output feature

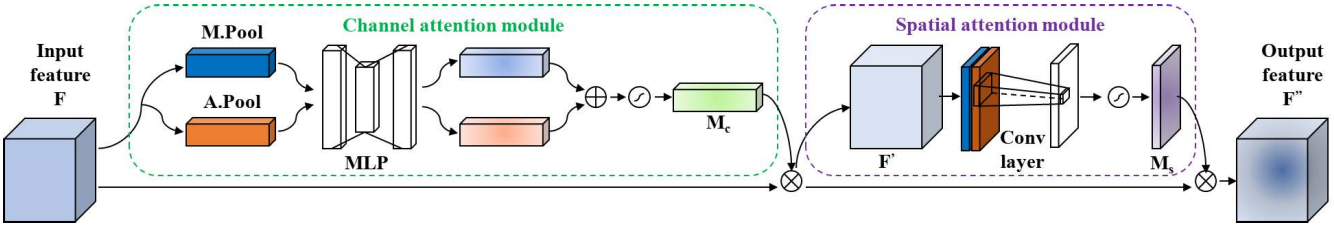

FIGURE 5. The structure of the convolution block attention module. It consists of two sub-modules: channel attention and spatial attention.

map of the adaptive connections. For each version:

$$\mathcal{F}_1'(x) = BN(f^{1\times1}(A.Pool(f^{3\times3}(x)))),$$
$$\mathcal{F}_2'(x) = BN(f^{1\times1}(f^{3\times3}(x))), \quad (2)$$

in which $f^{1\times1}$ and $f^{3\times3}$ are 1×1 and 3×3 standard convolution layers, respectively. *BN* is the batch normalization layer. *A.Pool* is the average pooling layer.

The $14\times14\times128$ feature map size (the output of Add 1) continues to pass through the CBAM [29]. This module helps the proposed network to focus on learning the salient information extracted from the previous module by the principle of channel attention and spatial attention. The output from the BAM is maintained at $14\times14\times128$ but the information of the object to be classified has been enriched to make it easier to distinguish from the background. The overall architecture of CBAM is depicted in Figure 5. The CBAM is composed of two sub-modules, channel attention and spatial attention. Suppose $F \in \mathbb{R}^{14\times14\times128}$ (output feature map from Add 3 with size $14\times14\times128$) is the input to the CBAM. It will sequentially generate channel attention map $M_c \in \mathbb{R}^{1\times1\times128}$ and spatial attention map $M_s \in \mathbb{R}^{14\times14\times1}$. The attention process is shown as follows:

$$F' = M_c(F) \otimes F,$$
$$F'' = M_s(F') \otimes F', \quad (3)$$

The symbol $\otimes$ represents the element-wise multiplication operation. F' and F'' present the intermediate and output feature maps, respectively. M_c and M_s are computed by the following equations:

$$M_c(F) = \sigma(MLP(A.Pool(F)) + MLP(M.Pool(F))),$$
$$M_s(F) = \sigma(f^{7\times7}(A.Pool(F) \parallel M.Pool(F))), \quad (4)$$

where σ describes the sigmoid function. The $\parallel$ symbol denotes the concatenation operation. *A.Pool*, *M.Pool*, $f^{7\times7}$, and *MLP* are the average pooling layer, the max pooling layer, a 7×7 standard convolution layer, and a multilayer perceptron with three hidden layers, respectively.

The last components in the feature extractor are two depthwise separable convolution layers with a similar kernel size of 3×3 and 256 and 10 channels, respectively. They act as transition modules between the feature extractor and classifier, making the network work quickly. Combined with the output of the fourth adaptive connection, they generate the final feature map with the dimensions of $14\times14\times10$ (number of channels corresponding to the number of classes to be classified).

B. CLASSIFIER MODULE

Traditionally, common classification networks have widely used fully connected layers at the end of the classification network. However, this technique significantly increases the network parameters, thus increasing the computational burden on the network and reducing the processing speed when applied on low computing devices. This study proposes a method to replace all fully connected layers in the popular classification networks with only one GAP layer. For this technique, the spatial features are extracted along each channel and the $14\times14\times10$ feature map from the extractor will quickly reduce the dimensions to $1\times1\times10$, saving a lot of network parameters. Finally, a softmax function is applied to calculate the probability of each object class appearing in the input image.

For simplicity, the categorical cross-entropy loss function is used to calculate the difference between the predicted value and the target value during training. It is defined as follows:

$$L_{cls} = -\sum_{i=0}^{9} p_i^*.log(p_i), \quad (5)$$

where i denotes the index of a class in the dataset. p_i^* is the target indicator which takes a value 0 or 1. p_i represents the probability of prediction from the network. *log* is a natural logarithm function.

C. VIDEO TESTING SYSTEM

Figure 6 describes the overall video testing system in detail (Testing stage). The system consists of input, the trained model, and the output. In which, the input is a set of videos with different resolutions including VGA, HD, FHD. The model is trained on the State Farm dataset and the stored weight file. The output is message text signals on the screen including prediction class, accuracy, and speed in FPS. This system can flexibly replace the input with a conventional camera, and the output can install audio signals to the speaker to alert the driver. This is the structure of the real-time driver warning system.

IV. EXPERIMENTS

A. DATASETS

There are three datasets for driver behavior classification used in this paper for the training and evaluation phases: the State Farm Distracted Driver Detection (State Farm) [18], the American University in Cairo version 1

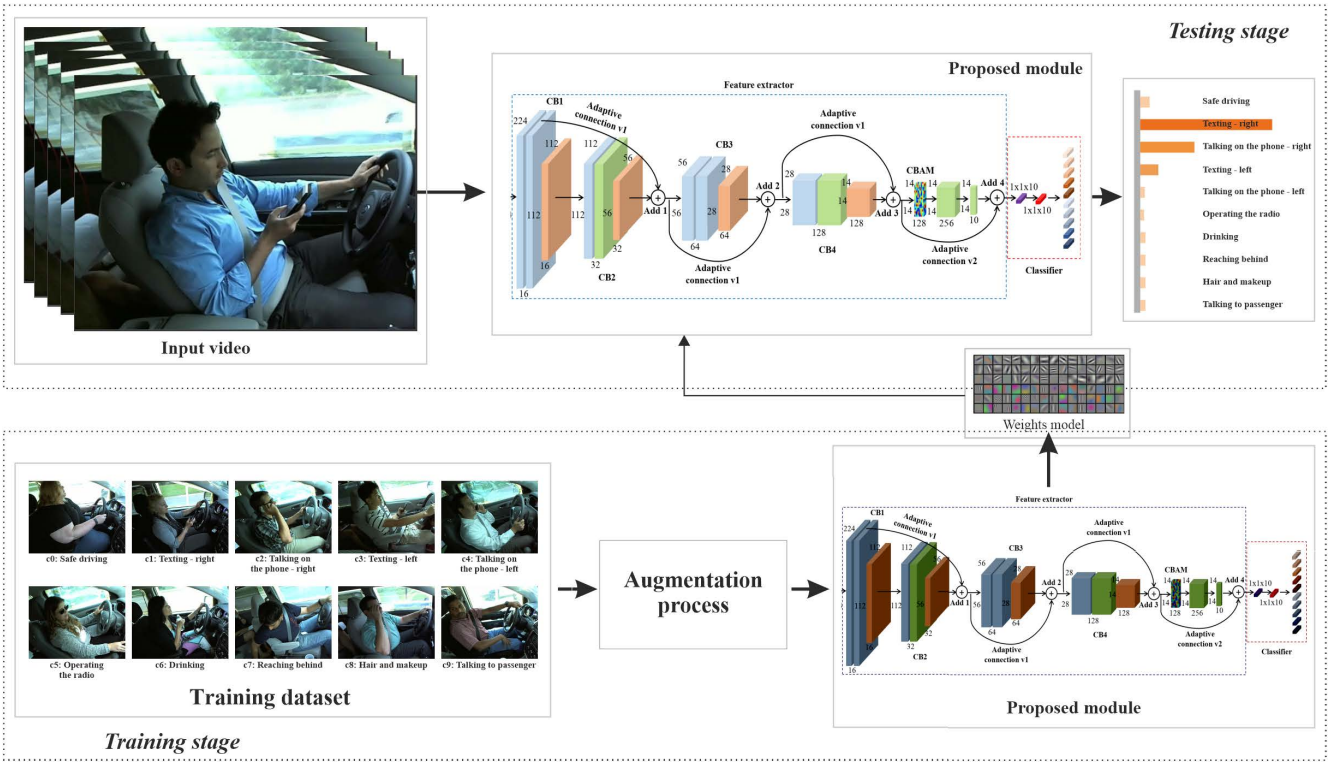

FIGURE 6. The overall video testing system.

(AUC version 1) [19], and the American University in Cairo version 2 (AUC version 2) [20].

1) STATE FARM DATASET

This dataset was downloaded from a contest on the Kaggle website. It is composed of 22,424 color images which have a resolution of 640 × 480 pixels. The images are separated into ten folders, each of which corresponds to one class, such as: safe driving (c0), texting - right (c1), talking on the phone - right (c2), texting - left (c3), talking on the phone - left (c4), operating the radio (c5), drinking (c6), reaching behind (c7), hair and makeup (c8), talking to passenger (c9).

2) AUC VERSION 1 DATASET

This dataset contains simulation driving behavior videos of 31 participants from seven countries with 22 males and nine females. The AUC version 1 dataset was taken by the SUS ZenPhone rear camera in video format. Then, 17,308 high-resolution images (1080 × 1920 pixels) were selected and processed. Following the division of the State Farm dataset, this dataset also has ten classes with similar labels (with different wording: Drive Safe, Text Right, Talk Right, Text Left, Talk Left, Adjust Radio, Drink, Reach Behind, Hair & Makeup, Talk Passenger) to the State Farm dataset.

3) AUC VERSION 2 DATASET

The AUC version 2 dataset was captured by two types of cameras, the ASUS ZenFone smartphone rear camera and the DS325 Sony DepthSense camera on five different brand cars in video format. The total number of participants is 44 people from seven countries with 15 females and 29 males. These videos were shot under various conditions such as lighting, driving conditions, and people wearing different types of clothing. A set of 14,478 images are extracted with a resolution of 1080 × 1920 or 640 × 480 pixels. The labels and corresponding folders of the classes are split like the AUC version 1 dataset.

B. EXPERIMENTAL SETTING

The proposed network is implemented using the Python programming language and the Keras framework. This network is trained as well as evaluated on a GPU (GeForce GTX 1080Ti). On the other hand, it was also used in a video testing system with VGA (640×480 pixels), HD (1280×720 pixels), and FHD (1920 × 1080 pixels) resolution videos on another CPU (Intel Core I7-4770 CPU @ 3.40 GHz, 32GB of RAM) and one Jetson Nano (Nvidia Maxwell GPU, 4GB of RAM). The training phase goes through 300 epochs with a batch size of 16. The Adam optimization method is applied for the weight update process of the network. The learning rate strategy is initialized by 10^{-3} then decreases gradually after 20 epochs with 0.55 times if the accuracy is not improved from the previous step. Each dataset is divided into a training set (80%) and an evaluation set (20%). To increase accuracy and avoid overfitting issues, this experiment uses several data

TABLE 1. The comparison results with different methods on the State Farm, AUC version 1, and AUC version 2 datasets. The red colored numbers represent the best competitors.

Model	Parameters (Million)	Acc (%)
State Farm dataset		
Simple CNN [24]	0.65	99.51
Light-weight CNN [24]	0.46	99.95
Mobile VGG [22]	2.20	99.75
Our network	**0.43**	**99.95**
AUC version 1 dataset		
VGG with Regularization [21]	140	96.31
Original VGG [21]	140	94.44
GA weighted ensemble [19]	120	95.98
Majority Voting ensemble [19]	120	95.77
AlexNet [19]	62	93.65
InceptionV3 [19]	24	95.17
Modified VGG [21]	15	95.54
DenseNet+Latent Pose [21]	8.06	94.20
NasNet Mobile [22]	5.30	94.69
MobileNet [22]	4.20	94.67
MobileNetV2 [22]	3.50	94.74
Mobile VGG [22]	2.20	95.24
SqueezeNet [22]	1.25	93.21
Light-weight CNN [25]	0.46	95.36
Our network	**0.43**	**95.57**
AUC version 2 dataset		
AlexNet [20]	62	94.29
InceptionV3 [20]	23.91	90.07
Resnet50 [20]	23.77	81.70
VGG16 [20]	15.25	76.13
Our network	**0.43**	**99.61**

augmentation methods such as random brightness, random zoom, and shift.

C. RESULT ANALYSIS

The proposed network was trained and evaluated on the three datasets mentioned above and tested on videos with different devices including a GPU, a CPU, and a Jetson Nano device. These experiments are reported based on the accuracy and frames per second (FPS) metric, respectively. As a result, this network achieved accuracies of 99.95% on the State Farm dataset, 95.57% on the AUC version 1 dataset, and 99.61% on the AUC version 2 dataset with only 426,785 parameters. In the usual way, the total network parameters are calculated based on the sum of all weights and biases of the convolutional and fully connected layers. In order to optimize network parameters, this paper has replaced four convolution layers with four depthwise separable convolution layers and wholly replaced the fully connected layers with a GAP layer. This dramatically reduces the network parameter but still ensures the feature extraction and classification accuracy of the network. The above result shows that, with the State Farm and AUC version 2, the results are almost absolute because the images in these datasets are clearly divided into folders following the class labels. In contrast, with the AUC version 1 dataset, several images lie in between the two behaviors, creating confusion in the learning process of the network. For the State Farm dataset, the proposed network outperforms the Simple CNN [24] and the Mobile VGG [22]. It is equivalent to the Light-weight CNN [24] but with nearly 30K fewer network parameters. For the AUC version 1 dataset, the proposed network outperforms most other networks and is only lower than the VGG with Regularization [21], the GA weighted ensemble [19], and the Majority Voting ensemble [24]. However, the network parameters of the VGG with Regularization is 325.58 times higher and the GA weighted ensemble and the Majority Voting ensemble are 279.07 times higher than proposed method. For the AUC version 2 dataset, the proposed network is completely outstanding in the popular classifier networks in [20] with a large difference in accuracy from 5.32% to 23.48%. Table 1 presents the accuracy comparison results of different networks on the three datasets. Figure 7 shows the qualitative classification results of proposed network on each dataset.

The three confusion matrices shown in Figure 8, Figure 9, and Figure 10 demonstrate the classification ability of the proposed network in each class. In the State Farm and the AUC version 2 datasets, the prediction rates of the classes are very uniform, ranging from 98% to 100%. In the AUC version 1 dataset, the prediction rates were mostly between 95% and 97% except for the "Adjust radio", "Reach behind", and "Hair & makeup" labels which were lower and ranged from 92% to 93%. From that observation, it can be seen that the driver's behavior is less related to other objects (cellphone, steering wheel, cup, bottle) and the ability to classify them is lower than other behaviors. This means the network is capable of focusing on the relevance of the different objects to make accurate classification decisions. This statement is proved with the Grad-cam visualization [30] in Figure 11.

For speed testing, this work also conducted the video testing system with the trained model on VGA, HD, and FHD resolution videos. The video inputs in this system can be replaced by a conventional camera with a speaker added for real-time deployment. However, in terms of driving safety, this work is only performed on real-time simulation videos. As shown in Figure 12, the network achieves the highest speeds of 243.78 FPS, 39.97 FPS, and 14.78 FPS on the GPU, the CPU, and the Jetson Nano device, respectively. Figure 12 also shows that when the video resolution changes from VGA to HD and FHD, the speed also decreases across all devices. This experiment also demonstrates the importance of choosing the right camera resolution for the system when applied in real-time systems. Therefore, it is recommended to use VGA resolution to ensure proper processing speed and avoid latency. With this result, the network can be deployed on low-computing devices and embedded systems to develop warning systems for vehicles.

As analyzed above, for the available image data, the proposed network works very well. However, when experimenting with videos or live-stream videos, sometimes the network showed several disadvantages. These include confusion between similar behaviors such as safe driving and texting when the driver's hand is close to the steering wheel, all-two-handed and one-handed driving behavior, hair-makeup, drinking, etc. The illumination conditions are also a factor

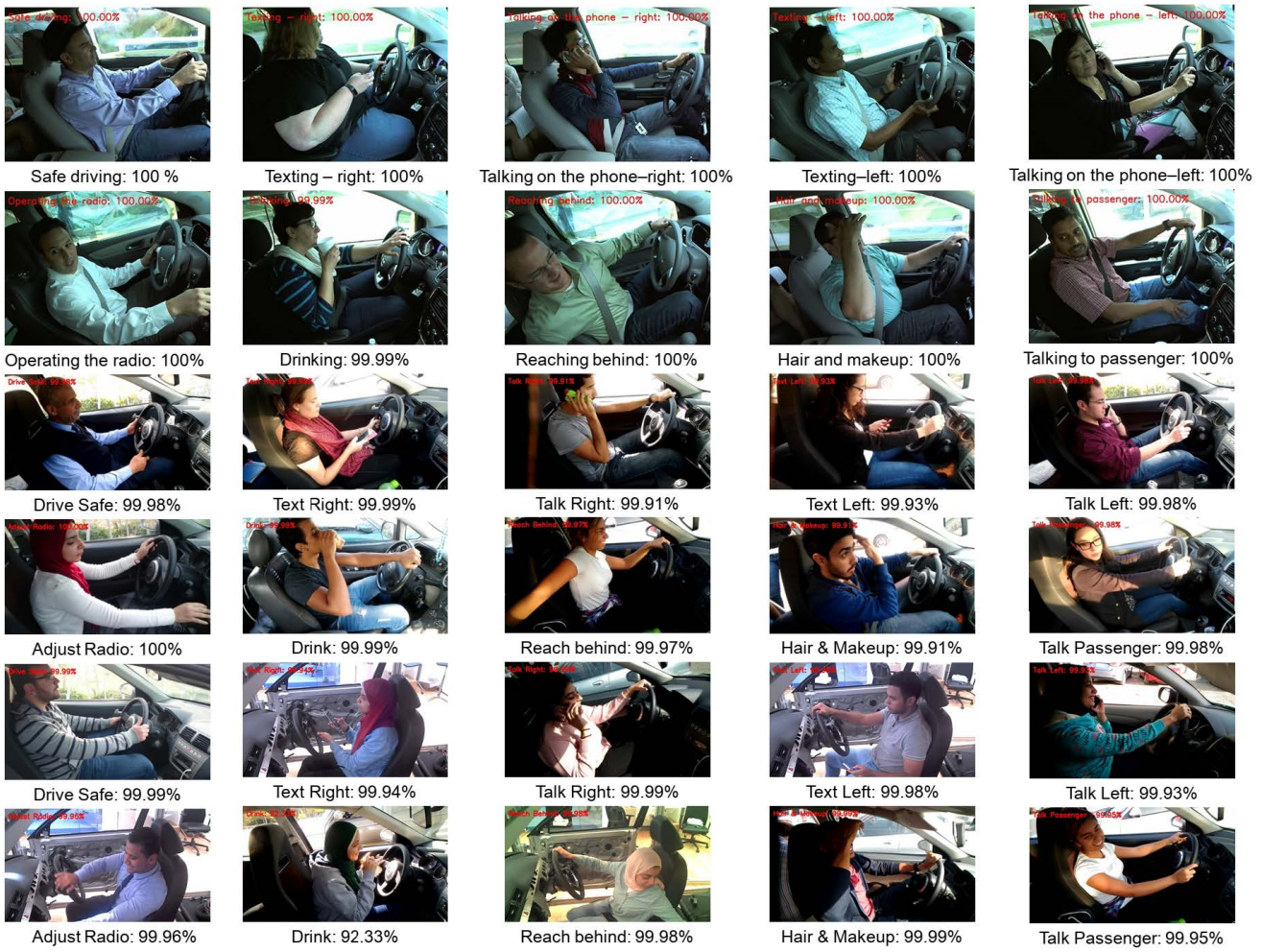

FIGURE 7. The qualitative results on the State Farm, AUC version 1, and AUC version 2 datasets. From top to bottom, the first two rows are for the State Farm dataset, the next two are for the AUC version 1 dataset, and the last two are for the AUC version 2 dataset.

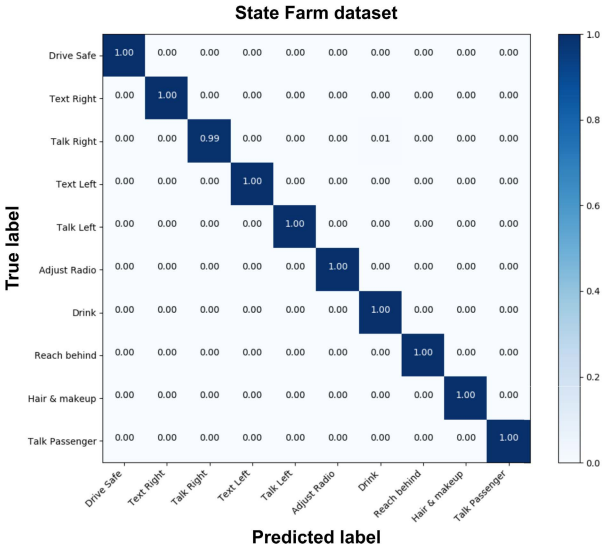

FIGURE 8. The confusion matrix on the state farm dataset.

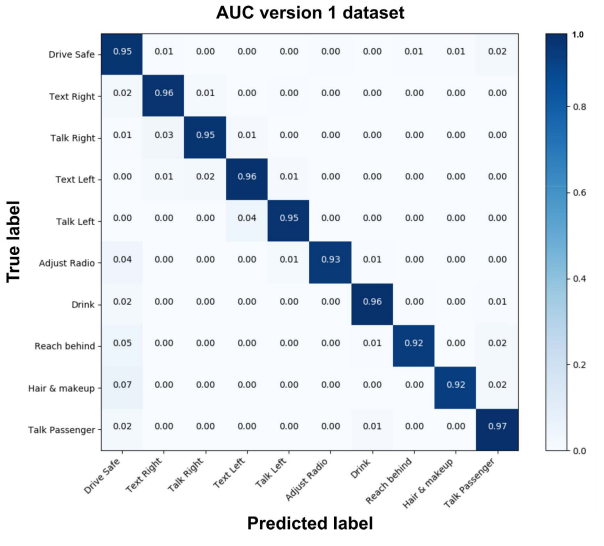

FIGURE 9. The confusion matrix on the AUC version 1 dataset.

that greatly affects the classification accuracy of the network. With too much light, the ability to distinguish foreground and background features will be reduced. In addition, the camera angle is also an important factor in determining the accuracy

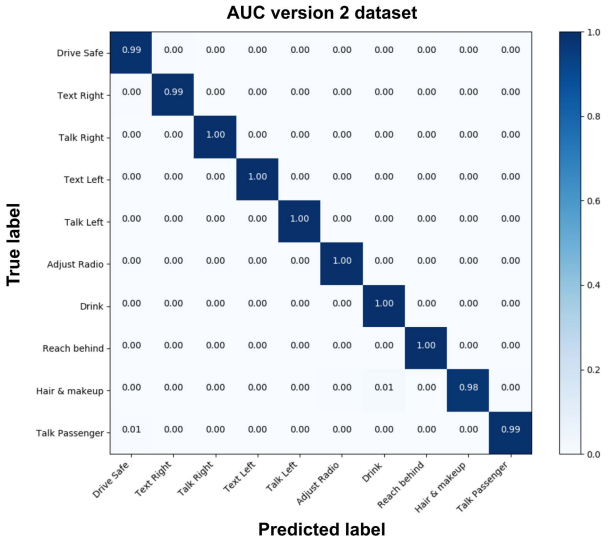

FIGURE 10. The confusion matrix on the AUC version 2 dataset.

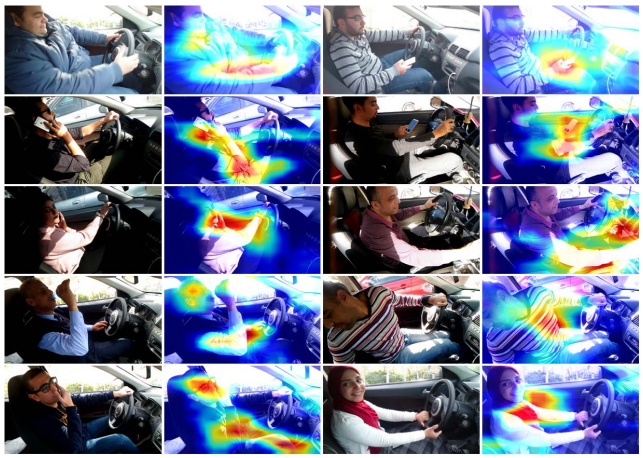

FIGURE 11. The grad-cam visualization on the AUC version 1 dataset. In each pair of images, the left one is the original image and the right one is a grad-cam visualization.

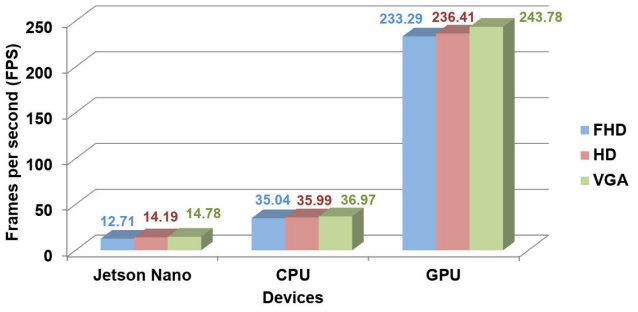

FIGURE 12. The speed of the proposed network with VGA, HD, and FHD video resolution on State Farm dataset and different devices.

of the network. If placed too far, the background area will increase making feature extraction difficult. If placed too close, the resulting image area is not large enough to clearly distinguish the driver's behavior. Therefore, it is a challenge to fine-tune the network and properly combine the above factors to improve that accuracy.

TABLE 2. Ablation study 1 of the proposed network on the AUC version 2 dataset. The red colored number represents the best competitor.

Modules	Network			
Stem	✓	✓	✓	✓
Adaptive connection			✓	✓
CBAM		✓		✓
Parameters	217,848	217,977	425,656	426,785
Accuracy (%)	98.58	98.61	98.59	**99.61**

TABLE 3. Ablation study 2 of the proposed network on the AUC version 2 dataset. The red colored number represents the best competitor.

Modules	Network		
Stem	✓	✓	✓
Adaptive connection	✓	✓	✓
CBAM			✓
BAM		✓	
SE	✓		
Parameters	426,687	426,774	426,785
Accuracy (%)	99.50	99.55	**99.61**

D. ABLATION STUDY

To evaluate the efficiency of each proposed module in the network, this experiment carried out several ablation studies. In ablation study 1, the stem module is first trained independently. Then, the stem, adaptive connection, and CBAM modules are combined one by one, and the results are compared with the stem module and the entire selection. The results in Table 2 show us that when only using the stem module, the accuracy is only 98.58% but the number of network parameters is only 217,484. The combination of the stem and CBAM modules increases the number of network parameters, and the accuracy is quite small. Combining the stem and adaptive connection modules nearly doubled the number of network parameters but increased accuracy by only 0.1%. Accordingly, when combining all three modules, the network achieves an accuracy of 99.61% with only 426,785 parameters. This experiment shows that the combination of the adaptive connections and CBAM significantly increases the accuracy. These are the core factors of the proposed network for achieving the best results. In another experiment for ablation study 2, the network architecture was trained and evaluated with different attention algorithms. Specifically, the network replaces the CBAM with a bottleneck attention module (BAM) [31] and an squeeze-and-excitation (SE) [32] respectively, and then compares the results. The obtained results are described in Table 3. Therefore, when replacing the CBAM with an SE, the number of network parameters is at least 426,687 and the accuracy is only 99.50%. When replacing the CBAM with a BAM, the accuracy increased by 0.5% and the number of network parameters also increased by 87 when compared with the previous result with an SE. Finally, the network using the CBAM. The number of network parameters only increased by 11 parameters as

compared to the BAM but reached the highest accuracy of 99.61%. This is the reason this paper chose the CBAM as the main attention algorithm to improve the classification accuracy for the proposed network.

V. CONCLUSION

This paper introduces a driver behavior recognizer based on a light-weight convolutional neural network and attention mechanism. This architecture exploits the advantages of standard convolution, depthwise separable convolution operation, and proposed adaptive connections to extract feature maps. Then, the network uses the CBAM attention mechanism to make the network focus on learning the most salient features. Finally, the classifier is applied to recognize ten driver behaviors. This work applied several techniques for reducing the number of network parameters and increasing the accuracy. The proposed network used all three benchmarks to train, evaluate, and report the results in the accuracy metric. On the other hand, it was also tested on different resolution videos with good processing speeds. In the future, this approach continues to develop based on a two-stage driver behavior warning system. The proposed network will be integrated into this system as the second stage after the driver body detection stage. By extracting the driver body positions first and then classifying it is possible to greatly increase the behaviors classification accuracy, especially with the real-time applications.

REFERENCES

[1] *Road Traffic Injuries*. Accessed: Jan. 25, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries
[2] *Distracted Driving*. Accessed: Jan. 25, 2022. [Online]. Available: https://www.nhtsa.gov/risky-driving/distracted-driving
[3] *Distracted Driving*. Accessed: Jan. 25, 2022. [Online]. Available: https://www.cdc.gov/transportationsafety/distracted-driving
[4] A. Eriksson and N. A. Stanton, "Takeover time in highly automated vehicles: Noncritical transitions to and from manual control," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 59, no. 4, pp. 689–705, 2017.
[5] H. M. Eraqi, M. N. Moustafa, and J. Honer, "End-to-end deep learning for steering autonomous vehicles considering temporal dependencies," *CoRR*, vol. abs/1710.03804, Dec. 2017, doi: 10.48550/arXiv.1710.03804.
[6] H. M. Eraqi, J. Honer, and S. Zuther, "Static free space detection with laser scanner using occupancy grid maps," *CoRR*, vol. abs/1801.00600, Jan. 2018, doi: 10.48550/arXiv.1801.00600.
[7] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 35–43.
[8] X. Zhang, N. Zheng, F. Wang, and Y. He, "Visual recognition of driver hand-held cell phone use based on hidden CRF," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Jul. 2011, pp. 248–251.
[9] R. A. Berri, A. G. Silva, R. S. Parpinelli, E. Girardi, and R. Arthur, "A pattern recognition system for detecting use of mobile phones while driving," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 2, Jan. 2014, pp. 411–418.
[10] Y. Artan, O. Bulan, R. P. Loce, and P. Paul, "Driver cell phone usage detection from HOV/HOT NIR images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 225–230.
[11] C. Craye and F. Karray, "Driver distraction detection and recognition using RGB-D sensor," 2015, *arXiv:1502.00250*.
[12] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 2953–2958.
[13] C. H. Zhao, B. L. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Intell. Transp. Syst.*, vol. 6, no. 2, pp. 161–168, 2012.
[14] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-RCNN approach to driver's cell-phone usage and hands on steering wheel detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 46–53.
[15] E. Ohn-Bar, "Driver hand activity analysis in naturalistic driving studies: Challenges, algorithms, and experimental studies," *J. Electron. Imag.*, vol. 22, p. 1119, Oct. 2013.
[16] C. H. Zhao, Y. S. Gao, J. He, and J. Lian, "Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier," *Eng. Appl. Artif. Intel.*, vol. 25, no. 8, pp. 1677–1686, 2012.
[17] C. Yan, B. Zhang, and F. Coenen, "Driving posture recognition by convolutional neural networks," in *Proc. 11th Int. Conf. Natural Comput. (ICNC)*, Aug. 2015, pp. 680–685.
[18] *State Farm Distracted Driver Detection*. Accessed: Jul. 14, 2021. [Online]. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection/data
[19] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," *CoRR*, vol. abs/1706.09498, Dec. 2018, doi: 10.48550/arXiv.1706.09498.
[20] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver distraction identification with an ensemble of convolutional neural networks," *J. Adv. Transp.*, vol. 2019, 2019.
[21] B. Baheti, S. Gajre, and S. Talbar, "Detection of distracted driver using convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1145–11456.
[22] B. Baheti, S. Talbar, and S. Gajre, "Towards computationally efficient and realtime distracted driver detection with MobileVGG network," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 4, pp. 565–574, Dec. 2020.
[23] A. Behera and A. H. Keidel, "Latent body-pose guided DenseNet for recognizing driver's fine-grained secondary activities," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.
[24] D. Tran, H. Manh Do, W. Sheng, H. Bai, and G. Chowdhary, "Real-time detection of distracted driving based on deep learning," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1210–1219, Dec. 2018.
[25] D.-L. Nguyen, M. D. Putro, and K.-H. Jo, "Distracted driver recognizer with simple and efficient convolutional neural network for real-time system," in *Proc. 21st Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2021, pp. 371–375.
[26] D.-L. Nguyen, M. Dwisnanto Putro, X.-T. Vo, and K.-H. Jo, "Light-weight convolutional neural network for distracted driver classification," in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2021, pp. 1–6.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2015, pp. 770–778.
[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2014, pp. 1–9.
[29] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[30] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that? Visual explanations from deep networks via gradient-based localization," *CoRR*, vol. 128, no. 2, pp. 336–359, Oct. 2016.
[31] J. Park, S. Woo, J. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," *CoRR*, vol. abs/1807.06514, Jul. 2018, doi: 10.48550/arXiv.1807.06514.
[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, pp. 7132–7141, Jun. 2018.

DUY-LINH NGUYEN (Member, IEEE) received the B.E. degree in applied informatics from the Vinh University of Technology Education, Vietnam, in 2010, and the master's degree in computer science from The University of Danang, Vietnam, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, South Korea. After the bachelor's degree, he joined the Department of Information Technology and Electrical Engineering, Quang Binh University, Vietnam, as a Lecturer. He worked with the Intelligent System Laboratory (ISLab), Department of Electrical, Electronic, and Computer Engineering, University of Ulsan. His research interests include object detection and recognition in computer vision based on machine learning.

MUHAMAD DWISNANTO PUTRO (Member, IEEE) received the B.Eng. (S.T.) degree in electrical engineering from Sam Ratulangi University, Manado, Indonesia, in 2010, and the M.Eng. degree from the Department of Electrical Engineering, Gadjah Mada University, Yogyakarta, Indonesia, in 2012. He is currently pursuing the Ph.D. degree with the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, South Korea. In 2013, he joined the Department of Electrical Engineering, Sam Ratulangi University, as an Assistant Professor. His current research interests include computer vision and deep learning, which focuses on robotic vision and perception.

KANG-HYUN JO (Senior Member, IEEE) received the Ph.D. degree in computer-controlled machinery from Osaka University, Osaka, Japan, in 1997. After a year of experience with ETRI as a Postdoctoral Research Fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea, where he is currently working as the Faculty Dean of the School of Electrical Engineering. His research interests include computer vision, robotics, autonomous vehicles, and ambient intelligence. He has worked as the Director or an AdCom Member of the Institute of Control, Robotics and Systems, The Society of Instrument and Control Engineers, and the IEEE IES Technical Committee on Human Factors Chair, an AdCom Member, and the Secretary, until 2019. He has also been involved in organizing many international conferences, such as the International Workshop on Frontiers of Computer Vision, the International Conference on Intelligent Computation, the International Conference on Industrial Technology, the International Conference on Human System Interactions, and the Annual Conference of the IEEE Industrial Electronics Society. He is currently an Editorial Board Member of international journals, such as the *International Journal of Control, Automation, and Systems* and *Transactions on Computational Collective Intelligence*.