Lightweight CNN-Based Driver Eye Status Surveillance for Smart Vehicles

Duy-Linh Nguyen, *Member, IEEE*, Muhamad Dwisnanto Putro, *Member, IEEE*, and Kang-Hyun Jo, *Senior Member, IEEE*

***Abstract*—Traffic accidents are the leading death rate among accident categories. One of the major causes of road traffic accidents is driver drowsiness. Many studies have paid attention to this issue and developed driver assistance tools to reduce the risk. These methods mainly analyze driver behavior, vehicle behavior, and driver physiology. This article proposes a driver eye status surveillance system based on lightweight convolutional neural networks (CNNs). The overall system consists of the following three stages: Face detection, eye detection, and eye classification. In the first stage, the system utilizes a small real-time face detector, named nano YOLO5Face. The second stage focuses on exploiting the compact CNN network architecture combined with the inception network, and triplet attention mechanism. Finally, the system uses a simple classification network architecture to classify open or closed eye status. Additionally, this work also provides the datasets for the eye detection task comprised of 10 659 images and 21 318 labels. As a result, the real-time testing reached 33.12 frames per second (FPS) and 25.11 FPS on an Intel Core I7-4770 CPU @ 3.40 GHz [personal computer (PC)] and a 128-core Nvidia Maxwell GPU (Jetson Nano device), respectively.**

***Index Terms*—Convolutional neural network (CNN), driver eye status surveillance system, eye classification, eye detection, face detection, triplet attention module.**

I. Introduction

ROAD traffic accidents claim more than one million people every year and 90% are caused by driver distraction [1]. Driving distraction can occur at any time while the vehicle is operating due to many factors [2]. Drowsiness is a common cause of distracted driving. Therefore, it is necessary to build driver assistance systems that can avoid accidents. A driver eye status surveillance system is an application built to alert drivers who are falling asleep. Drowsiness comes when the driver experiences a long journey, uses alcoholic drinks, such as alcohol and beer or has a medical condition. From those observations, the researchers mainly focused on analyzing driver behavior, vehicle behavior, and driver physiology [3]. Driver behavior can be recognized by extracting features of the driver body, such as the features of eyes, mouth, hands, head posture, and body posture. Vehicle behavior is a method of surveillance for unusual vehicle movements when going out of a lane, swerving on the road, or interacting with other vehicles in the vicinity. Driver physiology is monitored through electrical pulses from the heart rate, blood pressure, and changes in body temperature. The development of devices to monitor vehicle behavior and driver physiology require complex, high-cost devices, and synchronous infrastructure. Moreover, wearable devices can cause discomfort to the driver while operating and several natural factors in the driver body can interfere with the received electrical signal. Realizing the importance of eye status in the early stage [3] of sleep and the abovementioned analysis, this article proposes a driver eye status surveillance system. This system is based on lightweight CNNs with a new version of attention mechanisms named the triplet attention module. The use of lightweight CNN architectures to optimize network parameters while it is still ensuring the feature extraction process. On the other hand, the attention mechanism helps the system to focus on processing information in the eye area and ignores useless or background information. The proposed system aims to detect the status of the driver eye with the following three stages: face detection, eye detection, and eye classification. The system is built on the mechanism of driver eye status surveillance through the front-mounted camera. It does not influence the operation or does not cause unpleasant effects on the driver's body, as the abovementioned methods. The main contributions of this article are shown as follows.

1) This work proposes two lightweight CNNs for eye detection and eye classification tasks. These networks are aggregated with a real-time face detector (nano YOLO5Face) to build a three-stage driver eye status surveillance system. The CNNs in this article are designed for use in low-computing devices such as a CPU and an Nvidia Jetson Nano. With network parameter optimization based on compact architectural designs, the proposed networks solve the problem of computational and deployment costs. In addition, it is not invasive to the driver during use.

Manuscript received 16 March 2023; revised 9 June 2023; accepted 9 July 2023. Date of publication 17 August 2023; date of current version 23 February 2024. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the government (MSIT) under Grant 2020R1A2C200897212. Paper no. TII-23-0891. *(Corresponding author: Kang-Hyun Jo.)*

The authors are with the Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea (e-mail: ndlinh301@mail.ulsan.ac.kr; dputro@mail.ulsan.ac.kr; acejo@ulsan.ac.kr).

A set of demonstration videos and proposed datasets are available online at: https://bit.ly/3gEqnfY.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3296921.

Digital Object Identifier 10.1109/TII.2023.3296921

2) It also provides the datasets for locating the eye area in images under different situations and follows the PASCAL VOC dataset format. These are basic datasets for machine learning developers who use features from the driver eye. Along with the proposed eye detection and eye classification networks, they can use in other fields, such as eye-tracking, medical, and biomedical.
3) On the application side: the proposed system is tested in real-time on a CPU-based personal computer and a Jetson Nano device without high latency while ensuring accuracy.

The rest of this article is organized as follows. Section II introduces the related works in eye detection and classification including the advantages and disadvantages of each technical group. Section III explains in detail the proposed CNN architectures and real-time processing system. Section IV describes and analyzes the experiments and results. Finally, Section V concludes this article.

II. Related Works

A. Traditional Techniques

Traditional techniques relied heavily on the geometrical structure of the eye to detect the center eye position in an image or video. The work in [4] applied the Isophotes Curvature Estimation technique that determined the center of the eye and then built a voting system for that point. The authors in [5] introduced a method using an ensemble of randomized regression trees for locating the pupil of the eye. The method in [6] localized the center of the eye based on the image gradient and applied a squared dot product between the center candidates and the image gradient. The research in [7] used the Haar-like feature combined with a K-means cluster to navigate the pupil and fit an ellipse into the pupil using the Random Sample Consensus technique. The approach in [8] proposed an inner product detector based on correlation filters for detecting pupil landmarks. Traditional approaches applied computationally complex techniques and were very sensitive to environmental lighting conditions and facial occlusion, although they provided useful information.

B. Machine Learning Techniques

The eye is an important organ of the driver body and its status can reflect the first stage of drowsiness. From that observation, many machine learning studies have achieved high efficiency by focusing on exploiting and surveillance eye location. Several basic machine learning methods have been applied, such as the Haar wavelet and support vector machine (SVM) [9], histogram of oriented gradient-based SVM [10], self-similarity information combined with shape analysis [11], and Viola-Jones [12]. These techniques can overcome the disadvantages of traditional methods and achieve higher accuracy. However, they also need to be used in conjunction with other modern methods for implementation in real-time systems. Nowadays, the impressive advances of machine learning have made the development of computer vision applications based on CNNs more and more popular. In this trend, eye and pupil detection also used deep and complicated CNNs for improving extraction features and learning them. These CNNs were combined with multiple tasks to fully exploit the ability to learn important information from feature maps. Khan et al. [13] achieved 95% accuracy using the eyelid curvature angle as the basis for determining whether the eye is closed or open. The researchers in [14] considered drowsiness detection as an object detection task by identifying and detecting closed or open eyes. This work used a combination of MobileNet with single shot multibox detector (SSD) and achieved an average precision of 84%. The authors in [15] used the facial landmark technique to calculate the eye aspect ratio and eye closure ratio for drowsiness detection with 84% of accuracy. The study in [16] used an ensemble of two lightweight convolutional neural networks to extract and classify the eye patch with high accuracy. Another work [17] proposed a 4-D model for eye categorization by improving the VGG architecture family. In summary, machine learning methods exhibited outstanding advantages in feature extraction and eye location detection. In general, these methods focused on exploiting the features of the eyes and facial organs related to the sleepy state, such as the eyelids and mouth. Meanwhile, these organs were quite small in the image and lacked specialized datasets for detection tasks. It required complex techniques, high computational costs to accurately detect those locations, and high labor costs to annotate the datasets. Therefore, to simplify eye position detection and reduce the computational cost of the system, this article proposed an effective method of eye location detection and eye classification based on lightweight CNN architectures. Besides, this work also provides the datasets for eye position detection, which is popularly used in the object detection field.

III. Proposed Methodology

A. Eye Detection Network

The detailed architecture description of this network is shown in Fig. 1. It is built with four following modules: Shrinking, inception, triplet attention, and detection modules.

1) Shrinking Module: This module mainly uses 3×3 convolution layers and strides of 1, 2, 1, 2 in the convolution and max pooling layers, respectively. It helps to extract the basic features of the eye and reduce the input image from 128×128 to 32×32. In addition, the concatenated rectified linear unit (C.ReLu) module [18] is also applied to increase the efficiency of feature extraction. C.ReLu is described, as shown in Fig. 2(a).

2) Inception Module: The inception module is made up of six inception layers [19]. Each inception layer contains four convolution branches. Each branch uses one to three 1×1 or 3×3 convolution layers. Following each convolution layer is the batch normalization (BN) and the rectified linear unit (ReLU) activation function. In addition, the second branch adds one more max pooling layer to extract feature that is different from the remaining branches. With the idea of expanding the network width using multiple scales, this module increases the receptive field for the network. It maintains the scaling of the input feature map dimension 32×32 and provides output informative feature maps. Fig. 2(b) shows the structure of the inception layer.

3) Triplet Attention Module: The triplet attention module [20] consists of three branches that are shown in Fig. 2(c). It assumes that the input tensor is $F \in \mathbb{R}^{C \times H \times W}$ and passes it

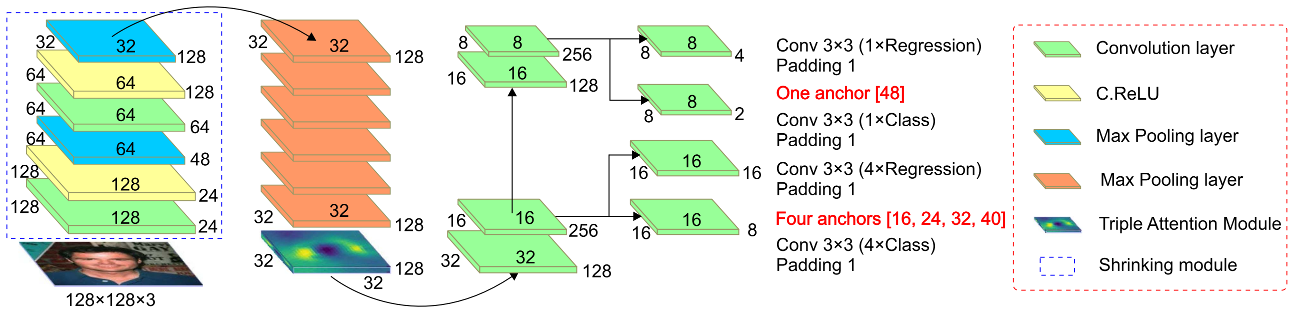
Conv 3×3 (1×Regression) Padding 1
One anchor [48]
Conv 3×3 (1×Class) Padding 1
Conv 3×3 (4×Regression) Padding 1
Four anchors [16, 24, 32, 40]
Conv 3×3 (4×Class) Padding 1
Convolution layer
C.ReLU
Max Pooling layer
Max Pooling layer
Triple Attention Module
Shrinking module
128×128×3

Fig. 1. Proposed eye detection network. It consists of shrinking, inception, triplet attention, and detection modules.

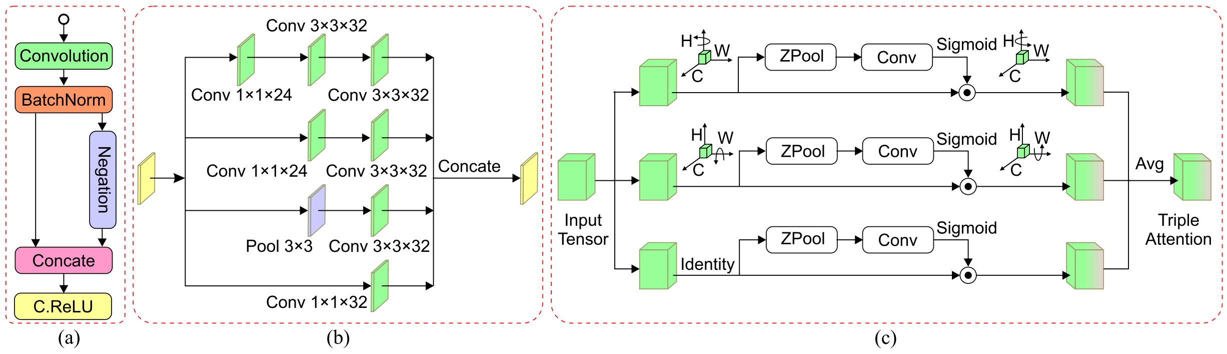

Fig. 2. Architecture of C.ReLU, (a) inception, (b) and triplet attention, (c) modules.

through each branch in this module. The first branch considers the interaction between the width dimension and the channel dimension of the feature map by applying a 90° anticlockwise tensor rotation along the H-axis and the rotated tensor annotated by $F_1^1 \in \mathbb{R}^{W\times H\times C}$. After that, F_1^1 go through the ZPool layer and the dimension of F_1^1 is reduced to $F_1^2 \in \mathbb{R}^{2\times H\times C}$. Then, F_1^2 is passed to a 1×1 convolution layer followed by the BN layer, which generates the output tensor of $(1 \times H \times C)$. This output goes through the sigmoid function σ to obtain the attention weights and applies them to the feature map F_1^1 using the channel element-wise multiplication operation. Then, the attention feature map rotates 90° clockwise along the H-axis to get the output feature map, which is the same shape as the input feature map F. Similarly, the second branch has the same architecture as the first branch except that it focuses on the relationship between the height dimension and the channel dimension. Hence, it applies a 90° tensor rotation to the W-axis. The last branch also uses the architecture of the first two branches but does not use tensor rotation. The results from the three branches were aggregated by simple averaging. Therefore, the final output of this module is an attention feature map of size $(C \times H \times W)$. The process of the attention module can be presented as

$$f_{\text{TA}} = \frac{1}{3}(\Re(F_1^1\sigma(\phi_1(F_1^2)) + \Re(F_2^1\sigma(\phi_2(F_2^2)) + F\sigma(\phi_3(F_3^2)) \tag{1}$$

where σ is sigmoid activation function; ϕ_1, ϕ_2, and ϕ_3 are 1×1 convolution layers followed by the BN layer. $\Re$ is 90° clockwise rotation to retain original input shape. The ZPool layer is presented as follows:

$$Z\text{Pool}(F) = [\mathcal{M}_{0d}(F), \mathcal{A}_{0d}(F)] \tag{2}$$

where $0d$ is 0th-dimension in which the max pooling ($\mathcal{M}$) and average pooling ($\mathcal{A}$) layers apply in the concatenation operation $[\cdot]$. The ZPool layer takes an input tensor of shape $(C \times H \times W)$ and generates an output tensor of shape $(2 \times H \times W)$. As can be seen, the triple attention module considers the input feature map tensor on all three dimensions, increasing the network's ability to focus on the prominent features of the human eye image.

4) Detection Module: First, this module uses four 1×1 and 3×3 convolution layers to further reduce the dimension of the feature maps and produce four feature maps of $32 \times 32 \times 128$, $16 \times 16 \times 256$, $16 \times 16 \times 128$, and $8 \times 8 \times 256$. From these output feature maps, the network only chooses two feature maps with dimensions of $16 \times 16 \times 256$ and $8 \times 8 \times 256$ as inputs for the detection module to detect eyes at two different levels. Detectors use two sibling convolution layers for classification and bounding box regression. To predict a bounding box, these detectors use different predefined square anchors of 16, 24, 32, 40 for small and medium eyes, and 48 for large eyes. The regression head generates a four-dimensional vector (x, y, w, h) as the offset of the bounding box location. The classification head generates a two-dimensional (eye or not-eye) vector as the label classification.

5) Loss Function: The loss function consists of classification loss and regression loss. The entire loss function is defined

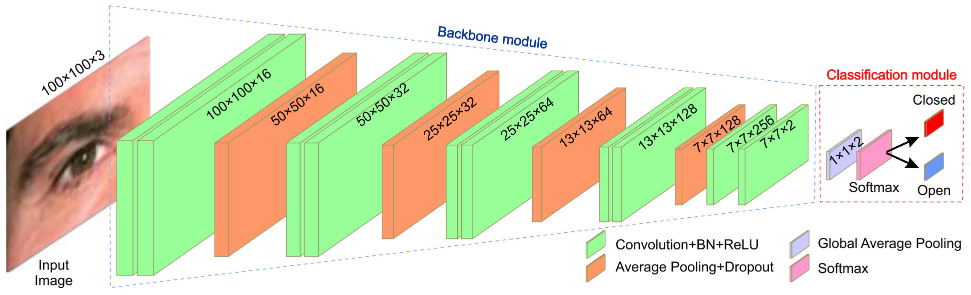

Fig. 3. Proposed eye classification network. It consists of two modules: backbone and classification.

as follows:

$$\mathcal{L}_{\mathrm{ed}}(p_i,t_i)=\frac{1}{N_c}\sum_i \mathcal{L}_c(p_i,p_i^*)+\lambda\frac{1}{N_r}\sum_i p_i^*\mathcal{L}_r(t_i,t_i^*) \quad (3)$$

where $\mathcal{L}_c(p_i,p_i^*)$ is the classification loss using the softmax-loss defined in (4), $\mathcal{L}_r(t_i,t_i^*)$ is the regression loss using the Smooth$L1$ loss defined in (5), p_i is the predicted probability and p_i^* is ground-truth label. t_i is the center coordinates (x,y) and dimension (height and width) of the predicted bounding box and t_i^* is the ground truth bounding box of object i. N_c is the mini-batch size, N_r is the number of anchor locations, and λ is the balancing parameter

$$\mathcal{L}_c(\mathcal{P}_i,\mathcal{P}_i^*)=-\sum_{i\in\mathrm{Pos}} x_i^p\log(\mathcal{P}_i)-\sum_{i\in\mathrm{Neg}}\log\left(\mathcal{P}_i^0\right) \quad (4)$$

where $x_i^p=\{0,1\}$ is the indicator for matching between the anchor box and ground-truth box of the object i. $\mathcal{P}_i$ and $\mathcal{P}_i^0$ are the probabilities for object and nonobject classification, respectively. Pos is a positive sample and Neg is a negative sample

$$\mathrm{Smooth}L1(x)=\begin{cases}0.5x^2, & \text{if } |x|<1\\ |x|-0.5, & \text{otherwise.}\end{cases} \quad (5)$$

B. Eye Classification Network

1) Network Architecture: The eye classification network is described in detail as shown in Fig. 3. This network is simply designed based on the basic usage of convolution, average pooling layers, and the softmax function to classify eye status. The eye classification network can be divided into two main modules: backbone and classification. The backbone module uses four convolution blocks and two separate convolution layers. Each of these blocks contains two standard convolution layers and one average pooling layer followed by BN and the ReLU activation function. The kernel sizes used in these convolution layers are 7×7, 5×5, and 3×3, respectively, according to the depth of the network. The feature extractor reduces the dimension of the input image from 100×100 to 7×7. In the classification module, this feature map is further reduced by the global average pooling layer to 1×1, with the two channels corresponding to the two classes in the dataset. Then, it applies the softmax function to calculate the probability of occurrence of the classes. The global average pooling layer, BN, and dropout techniques are employed to optimize network parameters and avoid overfitting issues.

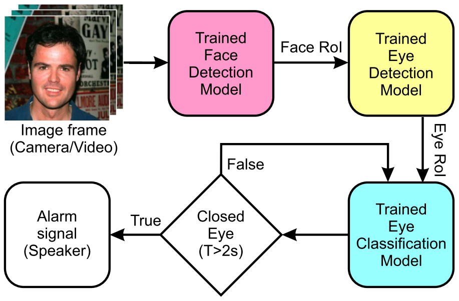
Image frame (Camera/Video)
Trained Face Detection Model
Face RoI
Trained Eye Detection Model
Eye RoI
Trained Eye Classification Model
Closed Eye (T>2s)
False
True
Alarm signal (Speaker)

Fig. 4. Block diagram of the proposed driver eye status surveillance system.

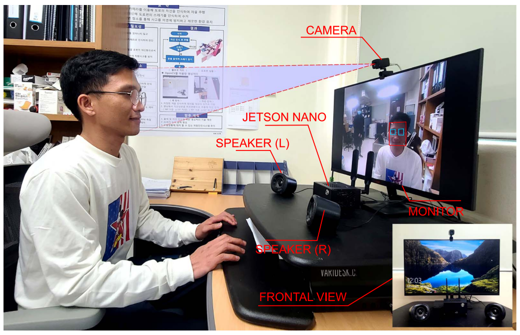

Fig. 5. Real-time testing setting with all devices.

2) Loss Function: The classifier uses the category cross-entropy loss function to compute the loss during training. This loss is shown as follows:

$$\mathcal{L}_{\mathrm{ec}}=-\sum_{i=0}^{1} t_i.\log(\mathcal{P}_i) \quad (6)$$

where i is the index of each class ($i=0$ to 1), t is the target indicator ($t=0$ or $t=1$), log is natural logarithm function, and $\mathcal{P}$ is the predicted probability of class i.

C. Real-Time Testing System

After training and evaluating each individual network on the respective datasets, the networks are recombined into a complete system and tested in real time, as described by the diagram in Figs. 4 and 5. The overall proposed driver eye status surveillance system consists of a face detection network, an eye detection network, an eye classification network, a camera (TGCAM-2000STAR), and two mini speakers. First, the nano YOLO5Face [21] network will detect the face areas in the videos. The YOLO5Face achieves state-of-the-art performance in the WiderFace dataset and the nano YOLO5Face is designed for real-time face detection on embedded or mobile devices. Then, these face areas are cropped and resized to 128 × 128 px as input to the eye detection network. In the next step, the eye detection network will detect the eye position in the previously cropped face areas, crop, and resize to 24 × 24 pixel. This is the most suitable size for eye areas chosen through many real-time tests. Finally, based on these cropped eye regions, the eye classification network will perform the classification of eyes with two statuses that are closed or open. If the closed eye exceeds the time threshold (2 s—This time can be set up by users), the speaker will sound a warning to the driver. The nonmaximum suppression (NMS) algorithm is also used in the first two stages to minimize the generation of redundant bounding boxes. The testing system uses the video graphics array (VGA) and high definition (HD) live-stream videos obtained from the camera connected to a PC (Intel Core I7-4770 CPU @ 3.40, 8 GB of RAM) and Jetson Nano device (128-core Nvidia Maxwell GPU, 4 GB of RAM), and the full high definition (FHD) videos captured from a cellphone and downloaded from YouTube. The distance from the camera to the participant's face is less than 0.5 m. The video resolutions used in this test were VGA (640 × 480 px), HD (1280 × 720 px), and FHD (1920 × 1080 px).

IV. Experiments

A. Dataset

Along with the design of the eye detection network, this article also proposes the eye datasets containing 10 659 images with 21 318 eye labels assigned according to the PASCAL VOC dataset format. The images in this dataset were collected from the BioID Face [22], CEW [23], GI4E [24], FEI Face dataset [25], and Yale Face Dataset B (YALEB) [26] dataset. The eye positions in BioID Face, CEW, and GI4E datasets are annotated automatically by the proposed Python program. Specifically, the procedure is to select 1521 gray-scale images with 384 × 286 pixel resolution from the BioID Face dataset. Each image shows the front view of the faces of 23 different people. From the center of the eye, the ground-truth bounding box is generated with a square size of 36 × 36. Similarly, the CEW dataset provides 2423 images with 100 × 100 pixel resolution, including 1192 images with both eyes closed and 1231 images with both eyes open. The ground-truth bounding box is also annotated from the center of the eye with a size of 24 × 24 px. The GI4E contains 1236 images from a standard camera with a resolution of 800 × 600 px in PNG format. Based on the position of the iris, the ground-truth bounding box is generated with a size of 46 × 46 px. The FEI Face database is collected from Brazilians aged 19 to 40 with distinct appearances, hairstyles, and makeup. The images in this dataset are taken on a homogenous background with up to 180° of rotation and a resolution of 640 × 480 px. This work selects 2800 images to annotate the eye areas with the LabelImg annotation tool. The YALEB is a quite large dataset for the face detection and recognition field. It contains 16 128 images of 28 people with 9 poses and 64 different illumination conditions. A set of 2679 images in this dataset were randomly selected and then the eye areas were also annotated using the LabelImg annotation tool. The LabelImg software is written in the Python programming language and presents a graphical interface in Qt. The bounding boxes of the eye positions are dynamically drawn based on the actual position of the eyes in the images. The location coordinates of eyes in each image are generated and saved in an XML file that follows the PASCAL VOC dataset format. Users can use these proposed datasets for different purposes to exploit the eye areas in the computer vision field.

For the eye classification network, the Closed Eyes In The Wild (CEW) [23] and MRL Eye [27] datasets are used for training and evaluation. The CEW dataset consists of 4846 images with 2384 closed eye labels and 2462 open eye labels. To enrich the dataset, during training, dataset augmentation methods are applied, such as vertical flip, contrast, and brightness changes. The MRL eye dataset contains 84 898 images collected from 37 different persons (33 males and 4 females) by three sensors (Intel RealSense RS 300, IDS Imaging sensor, and Aptina sensor). Two datasets are separated into 80% for training and 20% for evaluation.

B. Experimental Setup

1) Eye Detection Network: The eye detection network is implemented by the PyTorch framework and trained with 300 epochs on a GeForce GTX 1080Ti GPU, 32 GB of RAM. It is applied using several configurations, such as a batch size of 16, weight decay of 5×10^{-4}, momentum of 0.9, and the learning rate is initialized by 10^{-3} and descends to 10^{-5} for training. Stochastic gradient descent is also used to optimize the weight during back-propagation. The balancing parameter λ is set to 10. The NMS algorithm applies a threshold of 0.5.

2) Eye Classification Network: The eye classification network is built using the Keras framework and trained with basic settings for the classification task on a GeForce GTX 1080Ti GPU, with 32 GB of RAM. Specifically, the network goes through 200 epochs with the Adam optimization method and a batch size of 16. The learning rate is initialized to 10^{-4} and decreases after 10 epochs with a factor of 0.75 if the accuracy does not improve.

C. Experimental Results and Analysis

1) Eye Detection Network: This network plays an important role in determining the accuracy and efficiency of the entire eye status detection system. The eye detection network is trained and evaluated on proposed datasets based on BioID Face, CEW, GI4E, FEI Face, and the YALEB dataset. Additionally, to make a fair comparison with other network architectures, this work refines and retrains the FaceBoxes architecture [28], five variants of SSD architecture [29], nine variants of YOLO architecture [30], and the proposed network architectures with the change of attention modules (SE: Squeeze-and-Excitation,

TABLE I
COMPARISON RESULT OF EYE DETECTION NETWORK ON INDIVIDUAL PROPOSED DATASETS

Model	Input image	Parameters	GFLOPs	Average Precision (%)				
				GI4E	BioID	CEW	FEI	YALEB
Mobile architectures								
FaceBoxes [28]	128×128	844 610	0.40115	96.58	98.23	95.95	96.56	99.70
MobileNetV3-SSD [29]	300×300	2 141 372	0.52696	75.88	90.77	90.85	90.13	99.89
MobileNetV2-SSD [29]	300×300	3 051 928	0.65919	82.38	90.82	90.87	89.67	99.90
MobileNetV1-SSD [29]	300×300	3 990 680	1.24000	83.83	80.51	90.86	89.48	99.91
SSD300 [29]	300×300	23 745 908	62.82000	99.90	90.90	90.30	90.90	90.90
SSD512 [29]	512×512	23 745 908	180.61000	100	90.80	90.70	81.70	90.00
YOLOV3-Tiny [29]	416×416	8 669 876	12.90000	99.98	98.97	97.62	98.46	99.54
YOLOV3-SPP [30]	416×416	62 573 334	155.70000	100	99.00	97.97	98.32	99.93
YOLOV3 [30]	416×416	8 669 876	154.90000	**100**	97.90	**99.73**	99.40	**99.96**
YOLOV4-Tiny [30]	320×320	3 061 636	6.40000	99.90	**99.70**	99.40	99.70	99.70
YOLOV4 [30]	320×320	60 426 006	131.10000	99.70	99.60	99.20	**99.96**	99.96
YOLOV5s [30]	320×320	7 063 542	16.40000	99.70	99.60	99.60	99.60	99.60
YOLOV5m [30]	320×320	21 056 406	50.40000	99.70	99.60	99.30	99.60	99.70
YOLOV5l [30]	320×320	46 631 350	114.20000	99.70	99.60	99.30	99.60	99.60
YOLOV5x [30]	320×320	69 117 998	170.50000	99.70	99.60	99.30	99.60	99.60
Our architectures								
Proposed	**128×128**	**965 186**	**0.4934**	**98.12**	**98.35**	**96.50**	**97.14**	**99.66**
Our (SE)	128×128	967 070	0.4926	97.30	98.21	95.12	97.02	99.60
Our (BAM)	128×128	969 343	0.4948	98.05	99.05	95.04	97.01	99.80
Our (CBAM)	128×128	969 354	0.4926	90.84	97.31	94.78	97.10	99.68

SE: Squeeze-and-excitation, BAM: Bottleneck attention module, CBAM: Convolutional block attention module. The red color indicates the best competitor. The bold entities indicate our proposed

BAM: Bottleneck Attention Module, CBAM: Convolutional Block Attention Module). From the results in Table I, it demonstrates that this network has superior detection capabilities compared to FaceBoxes and SSD network architectures on the CEW and FEI datasets except for all YOLO network architectures with a 96.50% and 97.14% AP, respectively. For the GI4E dataset, the proposed network achieves 98.12% AP, and it surpasses FaceBoxes and other network architectures except for SSD300, SDD512, and all YOLO network architectures. However, the number of parameters of the original SSD, YOLOV3, YOLOV4, and YOLOV5m are 24.60, 8.98, 62.61, and 21.82 times larger than the proposed network architecture, respectively. For the BioID dataset, this experiment achieves an approximate AP when replacing the triplet attention module by BAM with an accuracy of 98.35% and 99.05% AP, respectively. However, the BAM has more than 4157 network parameters. It also outperforms the other compared network architectures except for YOLO network architectures. In the end, the proposed network reaches a 99.66% AP on the YALEB dataset. This result is comparable to the FaceBoxes network, outperforming both SSD300 and SSD512, and it is better than several YOLO network architectures. However, it is slightly weaker than the other proposed network architectures. The reason is the YALEB dataset contains a large number of black and white images with many different lighting conditions levels. This makes it difficult for lightweight or shallow architectures to distinguish small objects in the driver eye regions. In terms of computational complexity, the proposed network is only marginally larger than the FaceBoxes network at 0.09 GFLOPs, greatly smaller than the SSD and YOLO network families [from 13 times (YOLOV4-Tiny) to 336 times (SSD512)].

2) Eye Classification Network: The eye classification network achieves accuracies of 97.53% and 98.52% on the CEW and MRL eye datasets, respectively. It outperforms all networks with just 632 978 parameters and 0.0067 GFLOPs. Specifically, when compared with the AlexNet architecture, the parameters and computational complexity of AlexNet are 106 and 201 times larger but it only achieves an accuracy of 96.71% and 98.42%, which is lower than the proposed network 0.82% and 0.10% on the CEW and MRL Eye datasets, respectively. With the SqueezeNet architecture, the parameters and computational complexity of the proposed network are nearly 2.5 and 0.84 times larger, but the accuracy of the proposed network is nearly twice as large on the CEW dataset and 0.20% better on the MRL Eye dataset. Table II shows a detailed comparison of results between the proposed eye classification approach and popular classification networks.

TABLE II
COMPARISON RESULT OF THE EYE CLASSIFICATION NETWORK WITH POPULAR CLASSIFICATION NETWORKS ON CEW AND MRL EYE DATASETS

Network	Parameters	GFLOPs	Accuracy (%)	
			CEW	MRL Eye
SqueezeNet	256 818	0.0008	50.72	98.32
Proposed	**632 978**	**0.0067**	**97.53**	**98.52**
MobilleNetV2	3 571 778	0.0026	67.11	68.53
MobileNet	4 280 514	0.0021	82.27	81.80
VGG13	7 052 738	0.0011	96.29	98.33
DenseNet121	8 089 154	0.0021	79.59	87.50
VGG16	15 242 050	0.0011	94.33	97.47
LeNet	15 653 072	0.1869	96.70	98.08
VGG19	20 551 746	0.0011	94.12	97.18
Xception	22 961 706	0.0011	62.99	97.53
ResNet50	23 591 810	0.0042	94.85	97.50
InceptionV3	23 903 010	0.0042	76.60	79.38
AlexNet	67 375 938	1.3500	**96.71**	**98.42**

The red bold indicates the best competitor.
The bold entities indicate our proposed method.

3) Real-Time Analysis: The proposed real-time driver eye surveillance system is described in Section III-C. As a result, the system correctly predicts the eye status with different head poses and situations, such as wearing glasses, facemasks, hats, wearing

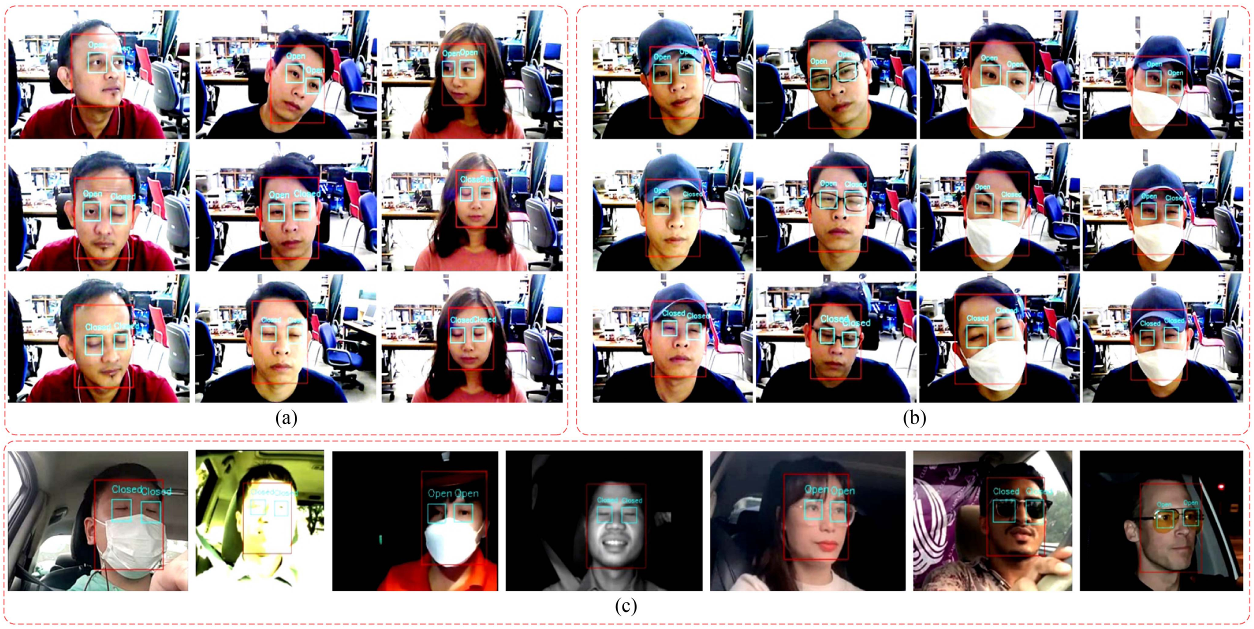

Fig. 6. Qualitative results of eye status surveillance system in VGA video live-stream on Jetson Nano device with three participants, (a) in VGA video live-stream on CPU-based PC with the single participant in different head pose and situations in the laboratory, (b) and in VGA video recorded in the car on CPU-based PC, (c) illumination changes (1–3), infrared video (4), myopia glasses (5), sunglasses (6), and the night vision glasses (7).

TABLE III
SPEED PERFORMANCE OF THE SYSTEM IN REAL-TIME TESTING ON PC AND JETSON NANO DEVICE

Device	Resolution	Face detection	Eye detection	Eye classification	Entire system
CPU	VGA	228.82	39.59	69.34	**33.12**
	HD	135.53	38.98	68.12	26.97
	FHD video	98.06	37.67	70.91	25.75
Jetson Nano	VGA	150.23	31.15	76.14	**25.11**
	HD	110.56	30.13	75.89	23.15
	FHD video	76.73	30.56	77.31	20.23

The red color indicates the best result. (Unit of measurement: FPS).
The bold entities indicate our proposed method.

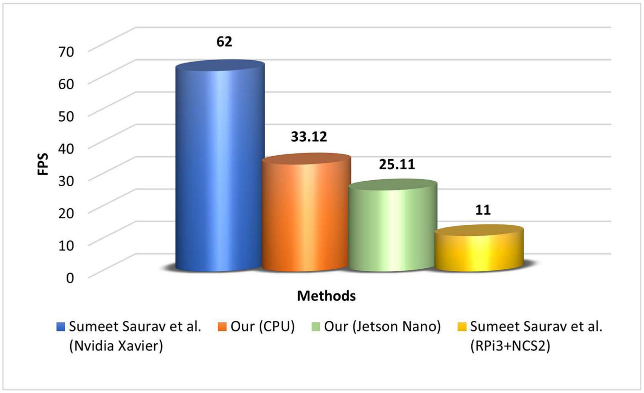

Fig. 7. Speed comparison between the proposed method and others in real-time testing on VGA (640×480) resolution.

both hat and facemask [see Fig. 6(a) and (b)], the changes of illumination, infrared video, and wearing different kind of glasses [see Fig. 6(c)]. The face detection network reaches up to 228.82 FPS on a CPU with VGA resolution and this is reduced to a minimum of 76.73 FPS on a Jetson Nano device with FHD resolution. Likewise, the eye detection network reaches 39.59 FPS on a CPU with VGA resolution and decreased to a minimum of 30.13 FPS on a Jetson Nano with HD resolution. In contrast, the eye classification network maintains speeds from 68.12 FPS on a CPU with HD resolution to 77.31 FPS on a Jetson Nano device with FHD resolution. The classification ability of the eye classification network on the Jetson Nano device is slightly better than that of the CPU. In total, the system reaches a minimum speed of 20.23 FPS on a Jetson Nano device with FHD resolution and a maximum speed of 33.12 FPS on a CPU with VGA resolution. The detailed execution speeds on the CPU and Jetson Nano devices are shown in Table III. This research also compares the speed of the proposed system and other methods in real-time testing. The experiment in [16] is conducted with two types of devices: Nvidia Xavier (512-core Volta GPU and 64 GB of RAM) and Raspberry Pi 3 (quad-core ARM Cortex-A53 CPU and 1 GB of RAM) with Intel Neural Compute Stick 2 (RPi3+NCS2). This work achieves 62 FPS and 11 FPS, respectively. The results in Table III and Fig. 7 prove that the speed of the proposed system is fall between the two above experiments and it works well in real-time with negligible delay to accurately predict the eye status in VGA resolution. The potential of the proposed system is that it can be deployed in real-time monitoring systems for drowsiness warnings in vehicles and in industry.

The proposed driver eye status surveillance system can be highly influenced by lighting conditions, which can significantly reduce the system's ability to detect eye areas. On the other hand, the distance and angle from the camera to the driver face is a factor that increases or decreases the size of the detected face and

TABLE IV
ABLATION STUDY OF EYE DETECTION NETWORK ON CEW DATASET

Modules	**Proposed network**				
Triple attention		✓	✓	✓	✓
Inception	✓		✓	✓	✓
C.ReLU	✓	✓		✓	✓
Detetors	2	2	2	3	2
Parameters (Million)	0.965	0.738	0.949	1.072	0.965
AP (%)	94.79	96.30	95.54	95.95	**96.50**

The red bold indicates the best competitor. The bold entities indicate our proposed method.

changes the accuracy. Besides, when the driver uses sunglasses, it is not possible to identify the eye areas because most of the eye area has been obscured. In this case, the system cannot recognize eye locations or predict the closed status. However, when using night vision glasses for driving, the detection results are still very accurate even when driving at night. Technically, because the system uses three stages, the execution speed is affected and depends heavily on the face detection network. In addition, the ability of the eye detection network also determines most of this speed. Therefore, shortening the system down to only two stages for direct eye detection and classification is necessary. This technique will greatly enhance performance on real-time systems. Fig. 6(c) shows VGA video testing results with different participants and different conditions on the CPU.

4) Ablation Studies: For the proposed eye classification network, this experiment evaluates the influence of the optimization methods and GAP layer. The ablation study 1 replaces the Adam optimizer with the AdamW optimizer, the accuracies achieved on CEW and MRL Eye datasets are 97.50% and 98.44%, respectively. The experimental results show that Adam is better than AdamW in optimizing the proposed eye classification network. For ablation study 2, this work replaces the GAP with fully connected (FC) layers using only one hidden layer (1024 network nodes) for training and evaluation in the same setting. The eye classification network has accuracies of 96.29% and 98.34% on CEW and MRL Eye datasets, respectively. This result is worse than the eye classification network that uses GAP (1.24% of accuracy on the CEW dataset and 0.18% of accuracy on the MRL Eye dataset). In addition, FC also increases a lot of network parameters (103 426 parameters) and computational complexity (0.6588 GFLOPs). Therefore, the use of GAP in the proposed eye classification network contributes greatly to the optimization of network parameters and computational complexity, aiming to apply on low computational devices.

To evaluate the effectiveness of different components in the eye detection network, each module was omitted during training and evaluation, then the results with the entire eye detection network were compared. The results in Table IV demonstrate that the use of the triplet attention module is necessary to increase the prediction ability (adding 1.71% to the AP) and maintain the parameters of the entire network. The omission of the inception module reduces a lot of network parameters (0.227 million parameters) but the precision is not significantly reduced (by only 0.2% of the AP). Therefore, the inception module may be omitted when designing the network to optimize the number of parameters. The C.ReLU module increases the efficiency of the feature extraction so when omitted it can reduce the AP by nearly 1%. Finally, using more than two detectors does not increase predictability, on the other hand, it increases the number of network parameters.

V. CONCLUSION

This article proposes a three-stage driver eye status surveillance system that includes face detection, eye detection, and eye classification stages. The research builds a complete driver eye surveillance system that achieves 33.12 FPS on VGA resolution. In addition, this work also provides the eye detection dataset in various scenarios. They serve as a foundation for drowsiness warning applications in smart vehicles. In the future, a two-stage driver eye status surveillance system will be developed focusing on the eye detection network modification to directly detect and classify the eye status without the face detection phase. The new system is applied in the night driving environment with the infrared camera.

REFERENCES

[1] "Road traffic injuries," 2023. Accessed: Jan. 19, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] S. Ansari, F. Naghdy, and H. Du, "Human-machine shared driving: Challenges and future directions," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 499–519, Sep. 2022.

[3] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, "A survey on state-of-the-art drowsiness detection techniques," *IEEE Access*, vol. 7, pp. 61904–61919, 2019.

[4] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1785–1798, Sep. 2012.

[5] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, "Eye pupil localization with an ensemble of randomized trees," *Pattern Recognit.*, vol. 47, no. 2, pp. 578–587, 2014.

[6] F. Timm and E. Barth, "Accurate eye centre localisation by means of gradients," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2011, pp. 125–130.

[7] L. Świrski, A. Bulling, and N. Dodgson, "Robust real-time pupil tracking in highly off-axis images," in *Proc. Symp. Eye Tracking Res. Appl.*, 2012, pp. 173–176.

[8] G. M. Araujo, F. M. Ribeiro, E. A. Silva, and S. K. Goldenstein, "Fast eye localization without a face model using inner product detectors," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 1366–1370.

[9] S. Chen and C. Liu, "Eye detection using discriminatory Haar features and a new efficient SVM," *Image Vis. Comput.*, vol. 33, pp. 68–77, 2015.

[10] R. Sharma and A. Savakis, "Lean histogram of oriented gradients features for effective eye detection," *J. Electron. Imag.*, vol. 24, no. 6, 2015, Art. no. 063007.

[11] M. Leo, D. Cazzato, T. De Marco, and C. Distante, "Unsupervised approach for the accurate localization of the pupils in near-frontal facial images," *J. Electron. Imag.*, vol. 22, no. 3, 2013, Art. no. 033033.

[12] V. Chirra, U. S. Reddy, and V. KishoreKolli, "Deep CNN: A machine learning approach for driver drowsiness detection based on eye state," *Revue d'Intelligence Artificielle*, vol. 33, pp. 461–466, 2019.

[13] T. Khan et al., "Smart real-time video surveillance platform for drowsiness detection based on eyelid closure," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–9, 2019.

[14] M. F. Shakeel, N. A. Bajwa, A. M. Anwaar, A. Sohail, A. Khan, and H.-ur-Rashid, "Detecting driver drowsiness in real time through deep learning based object detection," in *Advances in Computational Intelligence*, I. Rojas, G. Joya, and A. Catala, Eds. Cham, Switzerland: Springer, 2019, pp. 283–296.

[15] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som, and H. A. Ameen, "Drowsiness detection system using eye aspect ratio technique," in *Proc. IEEE Student Conf. Res. Develop.*, 2020, pp. 448–452.

[16] S. Saurav, P. Gidde, R. Saini, and S. Singh, "Real-time eye state recognition using dual convolutional neural network ensemble," *J. Real-Time Image Process.*, vol. 19, no. 3, pp. 607–622, 2022.

[17] I. Jahan et al., “4D: A real-time driver drowsiness detector using deep learning,” *Electronics*, vol. 12, no. 1, 2023, Art. no. e235. [Online]. Available: http://dx.doi.org/10.3390/electronics12010235

[18] W. Shang, K. Sohn, D. Almeida, and H. Lee, “Understanding and improving convolutional neural networks via concatenated rectified linear units,” in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 2217–2225.

[19] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[20] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, “Rotate to attend: Convolutional triplet attention module,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3138–3147.

[21] D. Qi, W. Tan, Q. Yao, and J. Liu, “Yolo5Face: Why reinventing a face detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 228–244.

[22] “The BioID face database,” 2020. Accessed: Oct. 23, 2020. [Online]. Available: https://www.bioid.com/facedb

[23] F. Song, X. Tan, X. Liu, and S. Chen, “Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients,” *Pattern Recognit.*, vol. 47, pp. 2825–2838, 2014.

[24] “Gi4e - Gaze interaction for everybody,” 2020. Accessed: Oct. 23, 2020. [Online]. Available: http://www.unavarra.es/gi4e/databases?languageId=1

[25] C. Thomaz and G. Giraldi, “A new ranking method for principal components analysis and its application to face image analysis,” *Image Vis. Comput.*, vol. 28, pp. 902–913, 2010.

[26] A. Georghiades, P. Belhumeur, and D. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[27] R. Fusek, “Pupil localization using geodesic distance,” in *Proc. Int. Symp. Vis. Comput.*, 2018, pp. 433–444.

[28] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, “FaceBoxes: A CPU real-time face detector with high accuracy,” in *Proc. IEEE Int. Joint Conf. Biometrics*, 2017, pp. 1–9.

[29] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[30] G. J. et al., “ultralytics/yolov5: V3.1 - Bug fixes and performance improvements,” Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

Duy-Linh Nguyen (Member, IEEE) received the Bachelor of Engineering degree in applied informatics from the Vinh University of Technology Education, Vinh, Vietnam, in 2010, the master’s degree in computer science from The University of Danang, Da Nang, Vietnam, in 2014, and Ph.D. degree in electrical engineering from the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, Ulsan, South Korea, in 2023.

After the bachelor’s degree, he joined the Information Technology and Electrical Engineering Department, Quang Binh University (QBU), Quang Binh Province, Vietnam, as a Lecturer. He is 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, the M.Eng. degree in electrical engineering from the Department of Electrical Engineering, Gadjah Mada University, Yogyakarta, Indonesia, in 2012, and the Ph.D. degree from the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, Ulsan, South Korea, 2022.

In 2013, he joined the Department of Electrical Engineering, Sam Ratulangi University, Manado, Indonesia, 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.

Dr. Jo was 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 for international journals, such as the *International Journal of Control, Automation, and Systems* and *Transactions on Computational Collective Intelligence*.