IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 21, 2022, accepted March 5, 2022, date of publication March 10, 2022, date of current version March 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3158304

Facemask Wearing Alert System Based on Simple Architecture With Low-Computing Devices

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 The Covid-19 epidemic has been causing heavy losses to humanity in terms of population, economy, and political stability. To deal with outbreaks of the pandemic, countries have been racing to develop vaccines and issue many regulations for people in daily life. Wearing a facemask in public is mandatory and will be severely punished if violated. In addition to the above mandatory regulations, it is necessary to develop tools for early warning when the human does not wear the facemask in public places such as offices, schools, supermarkets, train stations, etc. This paper proposed a facemask wearing alert system based on a simple convolutional neural network (CNN) operating on low-computing devices. This system works in two stages: face detection and facemask classification. In the first stage, it uses a face detection network with the main benefit of convolution, separable depthwise convolution, and double detectors layer to extract face region of interest (RoI). Then, this image area will go through a facemask classification network that exploits the advantages of convolution, separable depthwise convolution, and skip connection layers to classify facemask wearing (Mask or NoMask). The proposed networks are trained and evaluated on benchmark datasets. Along with simple designs, optimizing network parameters without ignoring accuracy, the system works in real-time at 33.17 and 26.18 frames per second (FPS) on an Intel Core I7-4770 CPU @ 3.40 GHz (Personal Computer - PC) and a Nvidia Maxwell GPU (Jetson Nano device), respectively. The demo video can be found here https://bit.ly/3yUgb8f.

INDEX TERMS Convolutional neural network, Covid-19, low-computing devices, facemask wearing alert system.

I. INTRODUCTION

Covid-19 is a dangerous pandemic that originated in Wuhan, China. According to statistics of the World Health Organization as of December 23, 2021, the world has about 276,436,619 infections and 5,374,744 deaths from Covid-19, and the number is increasing day by day [1]. The appearance of many strains of SARS-CoV-2 like the Delta and Omicron strains are serious and fast-spreading. Covid-19 is considered the biggest pandemic that has happened to humans, affecting the economy, politics, and social life of most countries in the world. Some studies have shown that facemasks can prevent infection from the corona virus [2]. The World Health Organization also recommends that people must wear a facemask when they have any symptoms of respiratory [3].

The governments of many countries have introduced many initiatives and mandatory regulations for people, especially the wearing of facemask in public places to limit the infection through the air. However, compliance with the law in some individuals is still limited, leading to the fact that not wearing a facemask can spread the disease. In addition, the management of wearing facemasks in public places with many people is difficult for the authorities because of the manpower, workers, and the risk of Covid-19 spreading among them. For poor countries, it is also much more difficult to deploy technical equipment to prevent the spread of the virus. Therefore, developing a toolkit for the automatic early alert of facemask wearing is essential. With the outbreak of CNN architectures in machine learning, many applications have been deployed to detect and localize facemasks in images, but the application in practice is still a challenge. Especially when deployed on low-computation devices such as CPU

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

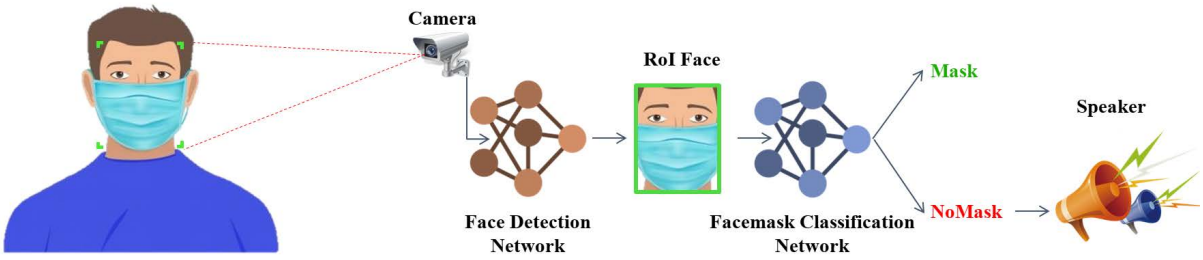

FIGURE 1. Overview of the proposed facemask wearing alert system. It consists of two stages: face detection and facemask classification.

and Jetson Nano, they require optimization of many factors to help the system operate smoothly with high accuracy. From those analyses, this paper focuses on researching and designing simple convolutional neural networks with few network parameters to build a facemask wearing alert system in public places. This design can take advantage of existing devices such as PC/laptop/Jetson Nano devices, cameras, and speakers without any additional deployment cost. The core contributions of this research are listed as follows:

1) Proposed simple and lightweight convolutional neural network architectures to build the facemask wearing alert system. This is a two-stage system that consists of face detection and facemask classification. These network architectures are used in the system to exploit the benefits of the convolution, separable depthwise convolution, and residual layers while maintaining accuracy.

2) Fully developed facemask wearing alert system that is easily deployed and used with existing equipment to detect facemask wearing with high accuracy.

The rest of this paper is presented as follows: Section II introduces facemask detection methods and their pros and cons. Section III details the architecture of the proposed convolutional neural networks used to build the system. Section IV analyzes and interprets the experimental results. The last part includes the conclusion and future works.

II. RELATED WORK

When the Covid-19 pandemic emerged and broke out, there have been many research studies focusing on detecting and classifying the wearing of facemasks. These methods can be separated into two groups, traditional methodology and CNN-Based methodology.

A. TRADITIONAL METHODOLOGY

These studies mainly apply traditional methods to extract facial features and then perform classification by simple approaches. The authors in [4] use the Viola-Jones algorithm to detect the human face in the images and then apply the PCA (Principal Component Analysis) algorithm to extract the facemask feature and classify the image as a facemask or a non-facemask. The method proposed in [5] uses the OpenCV library for preliminary face detection and the dlib library for facial feature extraction. Finally, they take advantage of the hierarchical framework and similarity algorithms for sample classification. Based on available machine learning library packages, [6] applies TensorFlow, Keras and OpenCV to directly detect facemasks appearing in images. These approaches are easy to implement but are still computationally complex due to the application of traditional algorithms. Therefore, they limit the abilities of feature extraction, detection, and classification.

B. CNN-BASED METHODOLOGY

The facemask wearing detection based on convolutional neural networks has been specially developed from the Covid-19 epidemic outbreak. These methods use deep learning network architectures to directly detect facemasks or to perform face detection combined with the facemask wearing classification. The work in [7] refines the InceptionV3 architecture to directly classify the non-masked faces. In another approach, [8] uses image processing methods to segment and extract features, then applies VGG-16 architecture to classify facemask wearing. This proposed system is tested on a Raspberry-pi device with an image test set. Method introduced in [9] uses a ResNet-50 backbone network to extract the feature maps and a YOLOV2 network to detect medical facemask. In the object detection field, [10] proposed a method that uses the RCNN network family (R-CNN, Fast R-CNN, and Faster R-CNN) and YOLOV2 to detect facemask and social distancing. The authors in [11] proposed a two-stage neural network architecture named SSDMNV2 which uses SSD network for face region detection and a MobileNetV2 network for facemask classification. Still exploiting the computational efficiency of the MobiletV2 network, [12] develops a masked detector deployed on embedded systems. [13] applies a hybrid between the transfer learning methods and machine learning methods to build the facemask detection system. Several recent studies implemented a real-time facemask detection system using the NVIDIA DeepStream SDK platform [14] and improved YOLOV4 [15]. Most of the above approaches have exploited modern convolutional neural network architectures for feature extraction and facemask detection. However, the application to real-time systems is limited due to the heavily weighted parameters, performed only on an individual object, only evaluated on image datasets, or only implemented on a GPU.

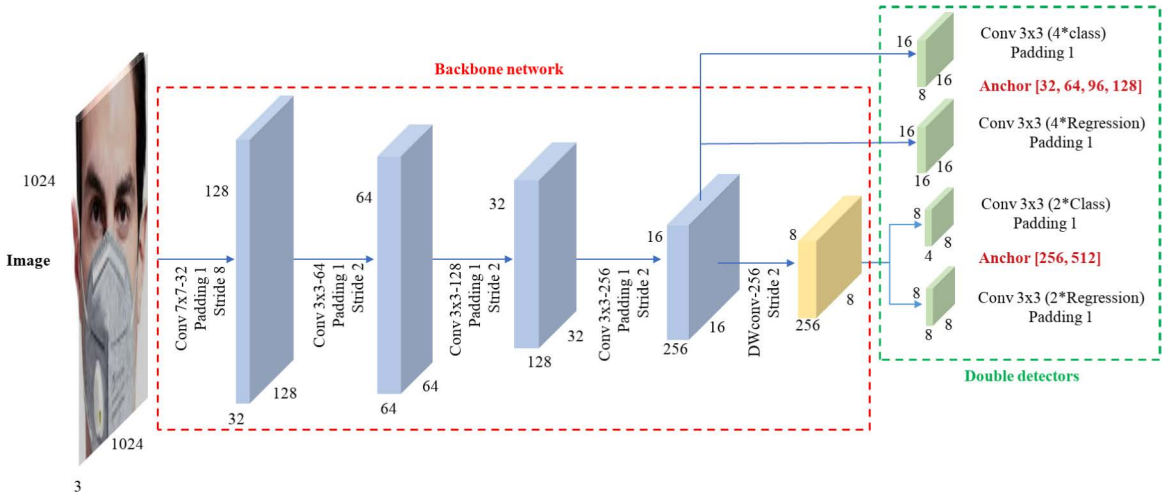

FIGURE 2. Structure of the proposed face detection network. It consists of a backbone network and double detectors.

III. PROPOSED METHOD

The entire facemask wearing alert system is detailed in Figure 1. The system is divided into two main stages, face detection and facemask classification. In the face detection stage, the paper proposes a simple convolutional neural network with double detectors for facial region extraction (RoI face) in multi-scale. This output is passed to the facemask classification stage which is a lightweight convolutional neural network to classify wearing a facemask or not. Additionally, this system is also integrated with a camera to capture images from public places and a speaker system to broadcast warning information.

A. FACE DETECTION NETWORK

The face detection network architecture is described in Figure 2. This network consists of a backbone network and double detectors.

1) BACKBONE NETWORK

This network is designed based on four convolution layers. First, a convolution layer uses kernel size 7×7 to quickly downsize the feature map. The first layer loses some information but still ensures the basic features of large and medium face sizes. The three sequential convolution layers all apply kernel size 3×3. The feature map will be reduced the size with a step of two over each layer. Following the last convolution layer is the depthwise separable convolution [16]. This convolution makes the detector work fast and save network parameters. With an input image of size 1024×1024 after going through the backbone network, a final feature map of size 8×8 is obtained. That means the backbone network reduces the input image dimension 128 times.

2) DOUBLE DETECTORS

To obtain the face RoI (region of interest) with multi-scale, this network uses double detectors. Each detector uses two 3×3 sibling convolution layers for classification and bounding box regression. Two layers take the last feature maps with sizes 16×16, and 8×8. The detectors use square anchor boxes of different sizes to predict the location of the face in the input image. For this work, it uses four anchor boxes with sizes 32, 64, 96, and 128 for small faces, one anchor box of size 256 for medium faces, and one anchor box of size 512 for large faces. Finally, each detector generates a four-dimensional vector (x, y, w, h) where (x, y) is the center coordinate, w is the width, and h is the height of the bounding box as the offset of the face position and a two-dimensional vector (face or non-face) for label classification.

3) LOSS FUNCTION

The face detection network uses MultiBox loss [17] to calculate the loss during training. The total loss is described as follows:

$$L_f\left(c_i, r_i\right) = \frac{2}{B}\sum_i L_{cls}\left(c_i, c_i^*\right) + \frac{1}{B}\sum_i L_{reg}\left(r_i, r_i^*\right), \quad (1)$$

where $L_{cls}\left(c_i, c_i^*\right)$ is the classification loss which using the softmax-loss shown as:

$$L_{cls}(c_i, c_i^*) = -\sum_{i \in Pos} x_i log(c_i) - \sum_{i \in Neg} log(c_i^0) \quad (2)$$

with x_i is an matching box indicator ($i-th$ anchor and ground truth), and c_i^0 is the confidence score of no object. $L_{reg}\left(r_i, r_i^*\right)$ is the regression loss defined as:

$$L_{reg}\left(r_i, r_i^*\right) = \sum_i H(r_i - r_i^*) \quad (3)$$

in which, H uses the *smooth* $L1$ loss to calculate:

$$H(x) = \begin{cases} 0.5x^2 & if \ \ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (4)$$

B is number of the matched boxes. c_i is the predicted label and c_i^* is the ground truth label of the $i - th$ anchor. r_i is the center coordinate (x, y) and dimension $(width, height)$ of the

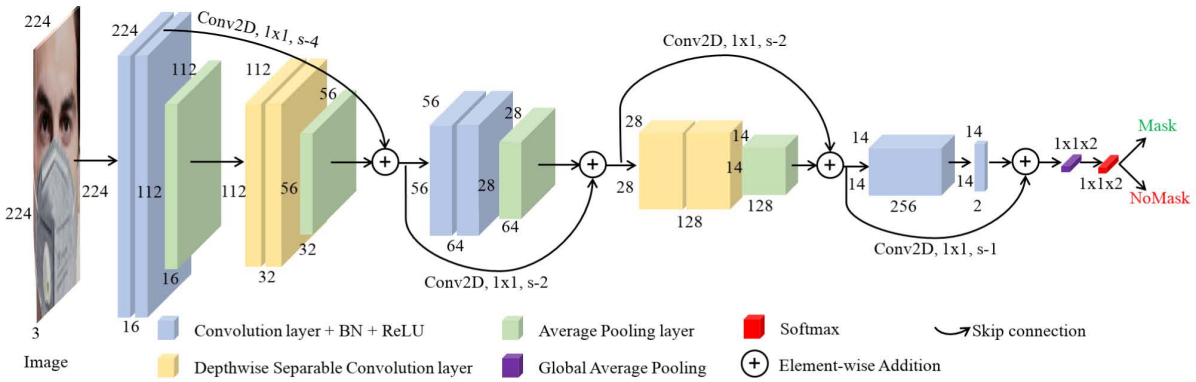

FIGURE 3. Structure of the proposed facemask classification network which consists of a stem, skip connection, and classification modules.

predicted bounding box and r_i^* is coordinate and dimension of the ground truth bounding box.

The bounding box regression process in this paper applies the parameterizations of four coordinates following [18] for predicted bounding box, anchor bounding box, and ground-truth bounding box. The equations of bounding box regression are shown as:

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a, \\ t_w &= log(w/w_a), t_h = log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a, \\ t_w^* &= log(w^*/w_a), t_h^* = log(h^*/h_a), \end{aligned} \tag{5}$$

where (x, y) denotes the center coordinate, w denotes the width and h denotes the height of the bounding box. The parameters x, x_a, and x^* are presented for the predicted bounding box, anchor bounding box, and ground-truth bounding box respectively (similar to the parameters in y, w, h).

B. FACEMASK CLASSIFICATION NETWORK

The details of the facemask classification network architecture are shown in Figure 3. This network includes three modules: stem, skip connection, and classification.

1) STEM MODULE

This module is built based on five convolution blocks comprised of two depthwise separable convolution blocks and three standard convolution blocks. Following each convolution block is an average pooling layer, except for the last one. The kernel size used in blocks decreases from 7×7 (first block) to 5×5 (second block) to 3×3 (third, fourth and fifth blocks). The convolution with a large kernel size at the beginning of the module is intended to increase the receptive field and capture the basic and useful information of the object that needs to be classified in the image. Otherwise, it is also the input of the first skip connection. Large kernels can increase the network parameters, but this problem has been overcome with interleaved depthwise separable convolution layers. The flexible calculation in depthwise separable convolution blocks helps to optimize network parameters

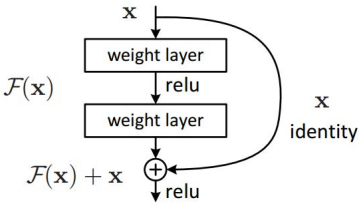

FIGURE 4. Structure of the skip connection layer [19].

and allows deployment on low-computing devices. The stem module servers like a multi-level feature map extractor to generate the feature maps with different scales. Thus, from the original image of size $224 \times 224 \times 3$, images go through this module and generate the final feature map of size $14 \times 14 \times 2$, where 2 is the number of classes.

2) SKIP CONNECTION MODULE

Going deeper in the neural network, the feature maps will lose large information. Therefore, combining current and previous level feature maps is necessary to maintain and enrich the amount of extracted information. This is the working principle of ResNet [19], as shown in Figure 4. Inspired by this network, the skip connection module uses four skip connection layers at four network levels: 56×56, 28×28, 14×14, and 14×14. The high-level feature maps combine with the lower-level features by using the element-wise addition operation to create a new feature map with the adaptive number of channels at each level of 32, 64, 128, and 256, respectively. Conducting the sequence on different levels helps the entire network to ensure the information extraction from end to end. In this case, each skip connection uses a 1×1 convolution layer followed by a batch normalization layer which is then combined with a higher level feature map.

3) CLASSIFICATION MODULE

The classification module is the last part of the facemask classification network. It composes two main layers, the global average pooling and the softmax layer. While most of the current popular classification networks use the fully connected and softmax layers to calculate the probabilities of classes, this network has been replaced by global average pooling

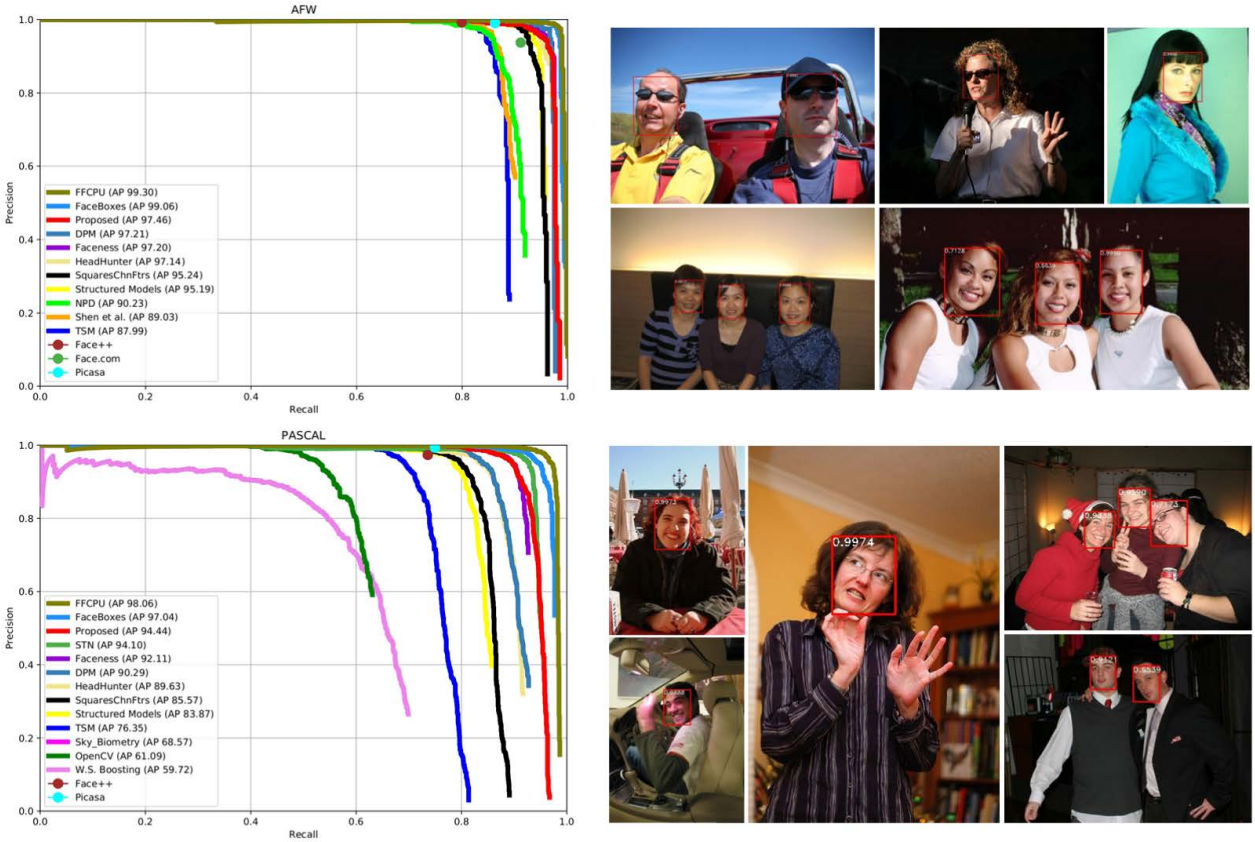

FIGURE 5. The evaluation and qualitative results of the face detection network on the AFW dataset (the first row) and the PASCAL FACE dataset (the second row). The number in each red box presents the confidence score of each bounding box.

and softmax layers. This has greatly reduced the number of parameters and still ensures classification accuracy.

4) LOSS FUNCTION

During training, the facemask classification network uses the categorical cross-entropy loss function to compute the loss. The categorical cross-entropy loss function is described in detail as the following equation:

$$L_c = -\sum_{i=1}^{2} t_i \cdot log(p_i), \tag{6}$$

where i is the number of classes (in this case, i set to 2), t is the target indicator ($t = 0$ or $t = 1$), p is the predicted probability, and log is the natural logarithm function.

C. REAL-TIME PROCESSING SYSTEM

After completing the training and evaluation of the separate networks on the respective datasets, two proposed networks are combined based on the description in Figure 1 for testing in the real-time system. The entire proposed facemask wearing alert system includes face detection network (using weight file was trained on the AFW dataset), facemask classification network (using weight file was trained on the CDD dataset), a camera (TGCAM-2000STAR), and a normal mini speaker. In the first stage, the face detection network will detect the face positions in the video obtained from the camera. The input video resolutions are the VGA (640 × 480 pixels) and HD (1280 × 720 pixels). The Non-Maximum Suppression (NMS) algorithm is used to reduce redundant bounding boxes and ensure to generate one bounding box which matches one face. After that, it crops the image areas containing only the face into separate regions and resizes them to 224 × 224 pixels. In the second stage, the facemask classification network will focus on previously cropped image areas to classify the face with a facemask or without a facemask. During the face classification process, if there is any face without a facemask, the system will generate an alert sound (in a beep) through the speaker.

IV. EXPERIMENTS

A. DATASET PREPARATION

1) FACE DETECTION DATASET

The face detection network uses the WIDER FACE dataset [20] to train and evaluates on two datasets: Annotated Faces in the Wild (AFW) [21] and PASCAL FACE [22]. The WIDER FACE dataset is a benchmark dataset that is more challenging than other datasets. It comprises 32,203 images with 393,703 face labels of various contexts such as scale, poses, and occlusions. All images are mainly selected from the WIDER dataset. The AFW includes 203 images with 473 labeled faces collected from the Flickr website. This dataset contains abundant backgrounds and challenges

TABLE 1. The comparison results of the facemask classification network in the accuracy (%) and FPS on Intel Core I7-4770 CPU @ 3.40 GHz (PC) with popular classification and other networks on four datasets. The "†" symbol presents the retrained results from the finetuned classification networks. The red color indicates the best competitor.

Model	# Parameters	SMFD	RMFD	FMLD	CDD	FPS
SqueezeNet †	256,818	**99.76**	99.03	**100**	97.31	25.59
Proposed	**413,616**	**99.52**	**99.83**	**100**	**99.02**	**39.88**
MobileNetV2 †	3,571,778	99.03	95.43	100	96.36	26.03
MobileNetV1 †	4,280,514	99.76	98.70	100	98.78	28.55
VGG13 †	4,758,978	98.79	98.90	100	**98.94**	8.69
NASNetMobile †	5,354,134	98.06	96.70	100	96.17	9.03
DenseNet121 †	8,089,154	99.76	99.13	100	98.29	11.21
VGG16 †	15,242,050	95.52	99.43	100	98.13	6.00
Xception †	15,242,050	99.52	99.47	100	98.29	5.97
VGG19 †	20,551,746	99.27	99.10	100	98.37	5.72
InceptionV3 †	23,903,010	99.52	92.97	100	96.58	9.28
ResNet50 †	25,687,938	98.82	**99.63**	100	97.96	7.55
LeNet †	78,428,072	97.56	99.13	100	95.44	20.96
Das et al. [6]	-	95.77	-	-	-	-
Loey et al. [13]	-	99.49	99.64	-	-	-
Chandrika Deb et al. [12]	-	-	-	-	98.00	-

FIGURE 6. The qualitative results of the facemask classification network on four datasets (the first row is the SMFD dataset, the second row is the RMFD dataset, the third row is the FMLD dataset, and the fourth row is the CDD dataset).

with different positions, expressions, and accessories. The PASCAL FACE dataset is a subset of the PASCAL VOC dataset. It contains 851 images with 1,335 labeled faces taken under various poses and backgrounds in indoor and outdoor conditions.

2) FACEMASK CLASSIFICATION DATASET

The facemask classification network was trained and evaluated on four datasets: Simulated Masked Face Dataset (SMFD) [23], Real-World Masked Face Dataset (RMFD) [24], Face Mask Lite Dataset (LMFD) [25], and Chandrika Deb's Dataset (CDD) [12]. The SMFD contains 1,376 images of which 690 are simulated facemask images and 686 are without facemask images. The RMFD consists of 5,000 images with facemask and 5,000 images without facemask. All images in the LMFD are produced using the Style GAN-2 network which is comprised of 10,000 HD images in each folder with facemasks and without facemasks. The last dataset is collected from Kaggle datasets, Bing search engine, and RMFD dataset. The CDD includes 4,095 images with 2,165 facemask images and 1,930 no facemask images. For a fair comparison with other methods, this work splits the datasets into 70% for the training phase and 30% for the evaluation phase.

B. EXPERIMENTAL SETUP

The experiments in this paper conduct training and evaluation on a GeForce GTX 1080Ti GPU with 32GB of RAM. On the other hand, for testing on a real-time system, this experiment uses one Intel Core I7- 4770 CPU @ 3.40 GHz with 32GB of RAM (PC) and one Nvidia Maxwell GPU with 4GB of RAM on Jetson Nano device connect to a TGCAM-2000Star camera.

1) FACE DETECTION NETWORK

The face detection network is trained with 300 epochs and several configurations are used with a batch size of 16, a weight decay of 5×10^{-4}, a momentum of 0.9, a learning

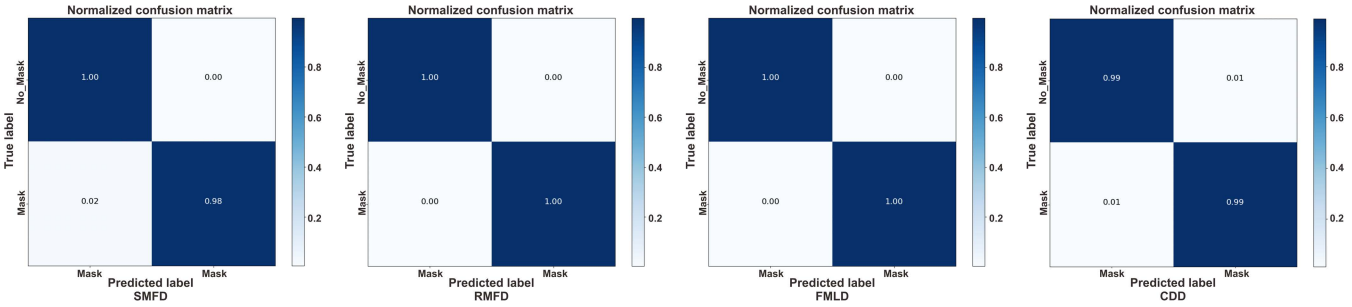

FIGURE 7. The confusion matrices for the SMFD, RMFD, FMLD, and CDD datasets.

rate from 10^{-6} to 10^{-3}, and the Stochastic Gradient Descent optimization method. To reduce the redundant bounding box, the detector uses an IoU (Intersection over Union) threshold of 0.5 in the Non-Maximum Suppression (NMS) algorithm.

2) FACEMASK CLASSIFICATION NETWORK

The facemask classification network is conducted using the Keras framework. It is trained with several basic configurations for the popular classification networks. In particular, this network undergoes the training of 200 epochs with Adam optimization and a batch size of 16. The learning rate is started from 10^{-4} and gradually decreases after 10 epochs with a coefficient of 0.75 if the accuracy does not increase.

C. EXPERIMENTAL RESULTS

1) FACE DETECTION NETWORK

The face detection network is trained on the WIDER FACE dataset and evaluated on two datasets, AFW and PASCAL FACE. As a result, this network achieves 97.46% and 94.44% of AP on the AFW and PASCAL FACE datasets, respectively. It outperforms traditional image processing and other CNN-based methods on both datasets. When compared to FaceBoxes [26] and FFCPU network [27] (the latest network architectures for face detection on CPU devices), its detection ability is weaker than the FaceBoxes and FFCPU architecture on the CEW and PASCAL FACE datasets. But it contains only 545,092 parameters, which is smaller than FaceBoxes network (844,610 parameters) by 1.55 times and FFCPU network (989,832 parameters) 1.82 times. Figure 5 shows the evaluation and qualitative results of the face detection network on the AFW and the PASCAL FACE datasets. From this result, it can be seen that the network has good ability to detect faces with different sizes and postures. In addition, it can also detect multiple faces appearing in a single image.

2) FACEMASK CLASSIFICATION NETWORK

The facemask classification network was trained and evaluated on the four datasets mentioned above and achieved accuracies of 95.52%, 99.83%, 98.94%, and 100% on SMFD, RMFD, CDD, and FMLD, respectively. Besides, this work selects the popular classification networks to finetune and retrain on the SMFD, RMFD, FMLD, and CDD datasets. To finetune these networks, all fully connected layers in each original network are replaced with a global average pooling layer to significantly reduce network parameters. As a result, this network outperforms popular classification networks, *Das et al. [6]* on SMFD, *Chandrika Deb et al. [12]* on CDD, and *Loey et al. [13]* on SMFD and RMFD datasets with only 413,616 parameters. On the SMFD dataset, it is only behind SqueezeNet in classification ability but the difference is only 0.24% and it outperforms other networks. The proposed facemask classification network also achieved outstanding speed compared to all other networks with 39.88 FPS on Intel Core I7-4770 CPU @ 3.40 GHz (PC) and 224 × 224 input size. Table 1 shows the comparison result of the facemask classification network in the accuracy (%) and FPS with popular classification and other networks on four datasets. The qualitative result of the facemask classification network on four datasets is presented in Figure 6. The confusion matrix in Figure 7 also demonstrates the balance in facemask classification ability in all four datasets of the proposed network.

TABLE 2. The speed (FPS) of the system in real-time testing on a PC and a Jetson Nano device with a single participant.

Device	Resolution	Face	Facemask	Overall system
PC	VGA	197.04	39.88	33.17
	HD	99.75	40.88	29.00
	FHD video	49.35	37.37	21.27
Jetson Nano	VGA	107.18	34.63	26.18
	HD	54.63	45.09	24.70
	FHD video	34.30	38.42	18.12

3) REAL-TIME ANALYSIS

The real-time system process is described in Section III. The whole system uses VGA (640 × 480 pixels) and HD (1280 × 720 pixels) live-stream videos obtained from a TGCAM-2000Star camera connected to a PC and a Jetson Nano device, and FHD (1920 × 1080 pixels) videos downloaded from YouTube. The experiments were carried out in a laboratory environment with the participation of male and female members. As the results show in Figure 8, the system accurately detects facemask wearing with different head postures and with several participants at the same time. The face detection network reaches 197.04 FPS with VGA resolution on the PC and gradually decreases to 34.30 FPS with FHD resolution on the Jetson Nano device. Meanwhile, the speeds of the facemask classification network are in the range from 34.63 FPS to 45.09 FPS. Therefore, the speed of the entire system also reaches 33.17 FPS with VGA resolution on the

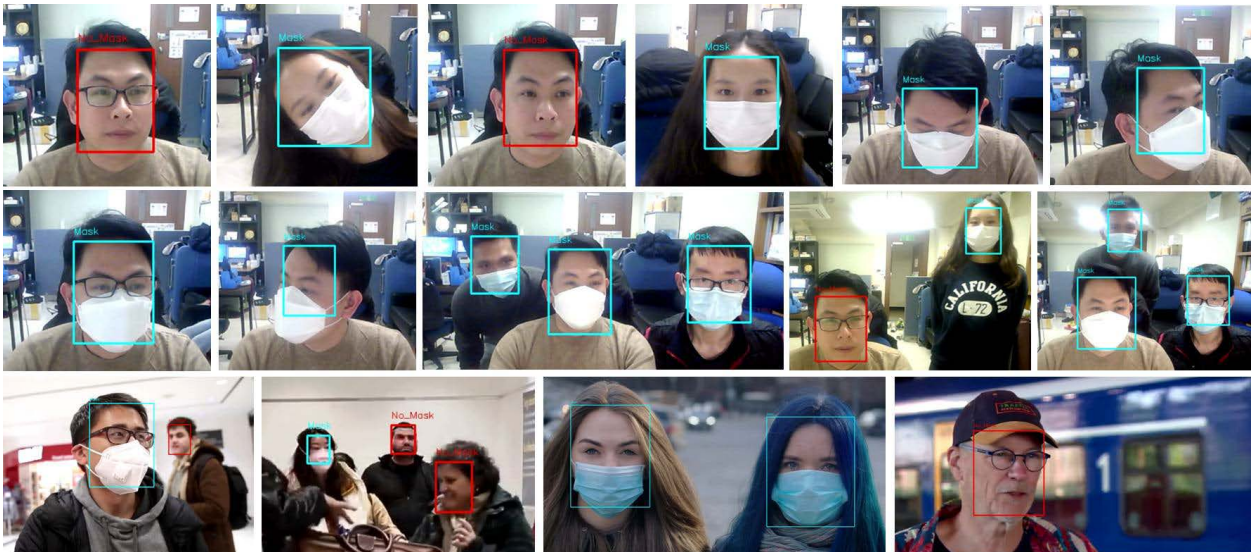

FIGURE 8. The qualitative results of the facemask wearing alert system in VGA video live-stream (two first rows) and FHD video from YouTube (third row) on a CPU-based PC.

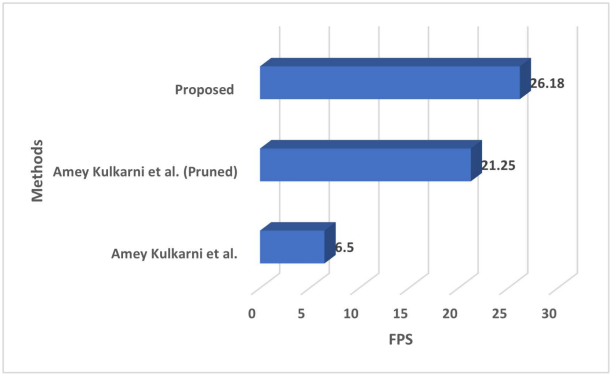

FIGURE 9. The speed comparison results (FPS) in real-time on the Jetson Nano device.

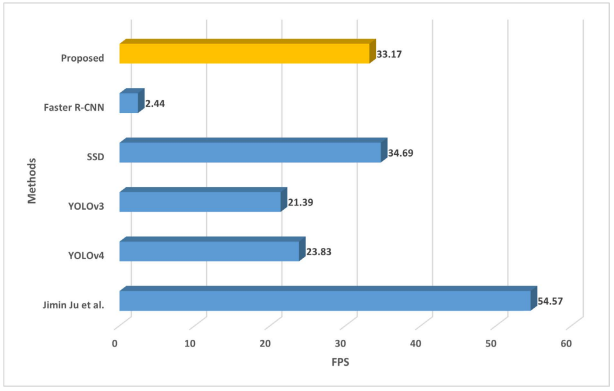

FIGURE 10. The speed comparison results (FPS) in real-time on the CPU and GPU devices. The yellow color presents the Intel Core I7-4770 CPU and the blue color presents the RTX 2070Super GPU.

PC and is reduced to 18.12 FPS with FHD resolution on the Jetson Nano device. The detailed system speed is shown in Table 2 proves that the proposed system can perform in real-time with negligible latency with low computing devices such as CPUs and Jetson Nano devices. On the other hand, this study also compared the performance in real-time with several experiments in [14] on the Jetson Nano device and found that the proposed method outperforms the methods in DeepStream SDK platform. When compared to the experiments in [15], the proposed method uses only the CPU but reaches speeds equivalent to SSD and outperforms Faster R-CNN, YOLOV3, and YOLOV4 on a GPU. It is only slower than the method proposed by *Jimin Ju et al.* (GPU-based method). Figure 9 and Figure 10 show the speed comparison results (FPS) in real-time testing.

During testing, this work found that the performance of the system may be affected by several environmental factors like illumination, camera quality, and especially the camera-object distance. In addition, the number of participants in a test also greatly affects the performance of the entire system. If the number of participants is crowded or exceeds the camera's frame will reduce the processing speed of the system. This paper tested and recommended that the number of people under three people ensures the operating speed and the system's stability. Table 3 shows the speed (FPS) of the system in real-time testing on a PC with VGA resolution and different participant numbers (one, two, and three participants).

TABLE 3. The speed (FPS) of the system in real-time testing on a PC with VGA resolution and the different participant numbers.

Participant numbers	FPS
One (Single)	33.17
Two	23.73
Three	15.08

4) ABLATION STUDY

The face detection is an important module in the facemask wearing alert system. To evaluate the face detection ability, this work tested three networks with three different configurations of the detector and anchor called single detection, double detection (proposed network), and triple detection.

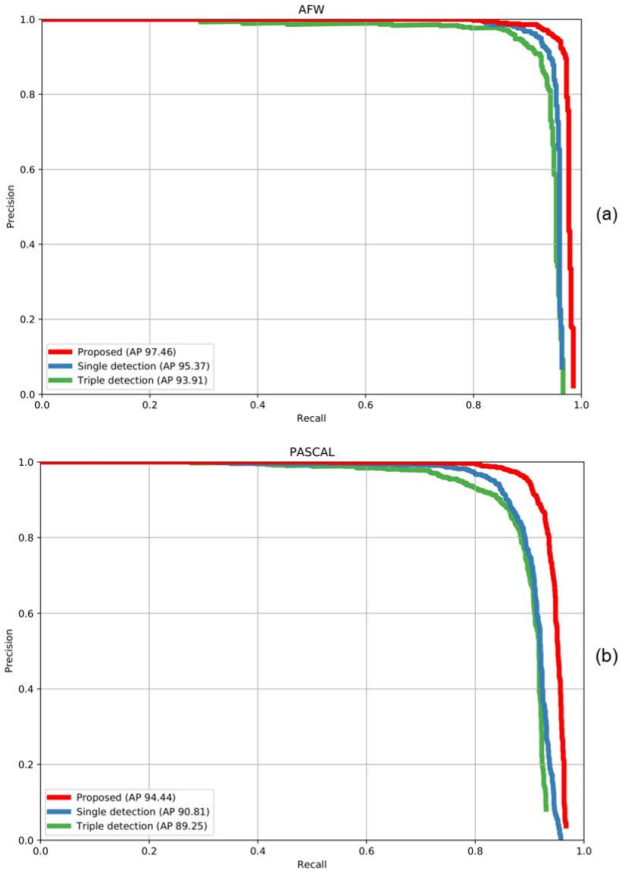

FIGURE 11. The comparison results in the face detection ability of three networks with different numbers of detectors.

The single detector uses all square anchor boxes of size 32, 64, 96, 128, 256, and 512 at 8 × 8 feature map, the triplet detector uses four square anchor boxes of size (32, 64, 96, 128) at 32 × 32 feature map, one square anchor box of size 256 at 16 × 16 feature map, and one square anchor box of size 512 at 8 × 8 feature map. The results in Figure 11 show that the single detection is quite weak because it only focuses on detecting large faces (at close range). It achieved 95.37% and 90.81% of AP on the AFW and PASCAL FACE datasets, respectively. In contrast, when increasing the number of detectors to three (triple detection), the detection ability decreased compared to single detection by 1.46% and 1.56% of AP, respectively. Additionally, the network increased by 13,830 parameters. With double detection, the number of parameters is between the other two networks but it can achieve the highest detection accuracy with 97.46% and 94.44% of AP on the AFW and PASCAL FACE datasets, respectively. Therefore, the proposed system used the network with double detection in the first phase to accurately detect the faces.

The facemask classification network is used in the last stage of the facemask wearing alert system. It is responsible for classifying facemask wearing and broadcasting alerts to speakers. This work also evaluates the classification ability of the proposed network through several designs. The results are shown in Table 4 prove that using fully connected layers does not increase the accuracy much (only 0.33%), but also significantly increases the network parameters (184,450 parameters) when compared to GAP. In addition, combining the fully connected layer and skip connection both increases the network parameters (up to 598,066 parameters) and reduces the accuracy (down to 97.80%). From the experiments, this study uses GAP combined with skip connection to achieve the best classification result of 99.02%.

TABLE 4. Ablation study of the facemask classification network on the CDD dataset. The red color presents the best result, FC denotes the fully connected layer, GAP denotes the global average pooling.

Modules	Network			
Skip connetion			✓	✓
FC	✓		✓	
GAP		✓		✓
Parameters	586,154	401,704	598,066	413,616
Accuracy (%)	98.70	98.37	97.80	**99.02**

V. CONCLUSION

This paper proposed a facemask wearing alert system based on simple and lightweight convolutional neural network architectures including face detection and facemask classification networks. The face detection network uses basic convolutional layers and double detection to detect faces in the scene. The facemask classification network is designed with convolution, depthwise separable convolutional layers, and the advantages of skip connection to classify faces with facemask and faces without facemask. If a face is without a facemask, the system will change the face bounding box color to the red color and play a sound to warn about not wearing a facemask in public places. With network parameter optimization, this system achieved up to 33.17 FPS and 26.18 FPS on CPU and Jetson Nano devices, respectively. The system can be deployed in low cost, available, and low-computing devices based on the CPU and edge devices. In the future, the system will be further developed to be able to detect faces in the far distance, very small size, and more people. The system will also be integrated with a social distancing alert system for dual duty.

REFERENCES

[1] *Who Coronavirus (COVID-19) Dashboard*. Accessed: Dec. 27, 2021. [Online]. Available: https://covid19.who.int/
[2] N. H. L. Leung, D. K. W. Chu, E. Y. C. Shiu, K.-H. Chan, J. J. McDevitt, B. J. P. Hau, H.-L. Yen, Y. Li, D. K. M. Ip, J. S. M. Peiris, W.-H. Seto, G. M. Leung, D. K. Milton, and B. J. Cowling, "Respiratory virus shedding in exhaled breath and efficacy of face masks," *Nature Med.*, vol. 26, no. 5, pp. 676–680, 2020.
[3] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, "Rational use of face masks in the COVID-19 pandemic," *Lancet Respiratory Med.*, vol. 8, no. 5, pp. 434–436, 2020.
[4] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of principal component analysis on masked and non-masked face recognition," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–5.
[5] Q. Chen and L. Sang, "Face-mask recognition for fraud prevention using Gaussian mixture model," *J. Vis. Commun. Image Represent.*, vol. 55, pp. 795–801, Aug. 2018.
[6] A. Das, M. W. Ansari, and R. Basak, "COVID-19 face mask detection using TensorFlow, keras and OpenCV," in *Proc. IEEE 17th India Council Int. Conf. (INDICON)*, Dec. 2020, pp. 1–5.

[7] G. J. Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face mask detection using transfer learning of InceptionV3," 2020, *arXiv:2009.08369*.

[8] S. V. Militante and N. V. Dionisio, "Real-time facemask recognition with alarm system using deep learning," in *Proc. 11th IEEE Control Syst. Graduate Res. Colloq. (ICSGRC)*, Aug. 2020, pp. 106–110.

[9] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustain. Cities Soc.*, vol. 65, Feb. 2021, Art. no. 102600.

[10] S. Meivel, K. I. Devi, S. U. Maheswari, and J. V. Menaka, "Real time data analysis of face mask detection and social distance measurement using MATLAB," *Mater. Today, Proc.*, Feb. 2021. [Online]. Available: https://www.sciencedirect.com/journal/materials-today-proceedings, doi: 10.1016/j.matpr.2020.12.1042.

[11] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustain. Cities Soc.*, vol. 66, Mar. 2021, Art. no. 102692.

[12] C. Deb. (2020). *Face-Mask-Detection*. [Online]. Available: https://github.com/chandrikadeb7/Face-Mask-Detection

[13] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, Jan. 2021, Art. no. 108288.

[14] A. Kulkarni, A. Vishwanath, and C. Shah. (2020). *Implementing a Real-Time, AI-Based, Face Mask Detector Application for COVID-19*. [Online]. Available: https://developer.nvidia.com/blog

[15] J. Yu and W. Zhang, "Face mask wearing detection algorithm based on improved YOLO-v4," *Sensors*, vol. 21, no. 9, p. 3263, May 2021.

[16] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," 2015, *arXiv:1512.02325*.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, *arXiv:1506.01497*.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[20] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," 2015, *arXiv:1511.06523*.

[21] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2879–2886.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and W. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.

[23] P. Bhandary. (2020). *Observations*. [Online]. Available: https://github.com/prajnasb/observations/

[24] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, H. Chen, Y. Miao, Z. Huang, and J. Liang, "Masked face recognition dataset and application," 2020, *arXiv:2003.09093*.

[25] P. Kottarathil. (2020). *Face Mask Lite Dataset*. [Online]. Available: https://www.kaggle.com/prasoonkottarathil/face-mask-lite-dataset

[26] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," 2017, *arXiv:1708.05234*.

[27] M. D. Putro, L. Kurnianggoro, and K.-H. Jo, "High performance and efficient real-time face detector on central processing unit based on convolutional neural network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4449–4457, Jul. 2021.

DUY-LINH NGUYEN (Member, IEEE) received the Bachelor of Engineering degree in applied informatics major 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 Information Technology and Electrical Engineering Department, Quang Binh University, Vietnam, as a Lecturer. He worked at 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 International Workshop on Frontiers of Computer Vision, International Conference on Intelligent Computation, International Conference on Industrial Technology, 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*.