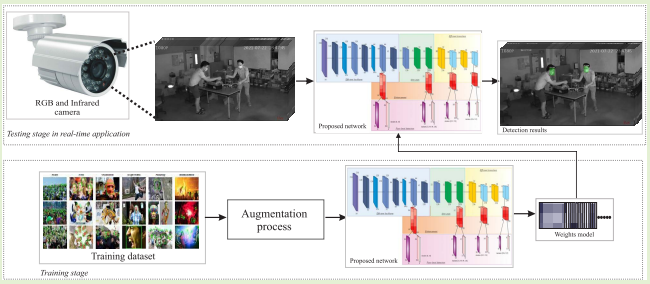
An Efficient Face Detector on a CPU Using Dual-Camera Sensors for Intelligent Surveillance Systems

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

***Abstract*—Intelligent surveillance systems require face detection to identify human facial areas. This system should be able to utilize dual-camera sensors (color/IR) for working every time. Additionally, practical application demands a detector to be operated in real-time on a low-cost device or CPU. The deep Convolutional Neural Network (DCNN) technique has successfully used a robust facial extractor, but it requires a high amount of computation for high-resolution input. On the other hand, the light architecture generates a large number of false positives as a fast detector. This feature extractor pays less attention to specific facial features and often ignores global and local relationships between elements. This paper proposes an efficient face detector to accurately localize faces using light architecture. The one-stage detector consists of an efficient backbone to rapidly extract features and a four-level detection layer to predict variations in facial scales. To improve the non-robust feature extractor, it implements an enhancement module to enhance specific facial features at each level without significantly increasing the parameters. The proposed detector uses knowledge from the WIDER FACE dataset to train the model with a gradual learning rate. The experiment results show the effectiveness of the detector in outperforming CPU-based detectors on benchmark datasets. It also runs in real-time at 27 frames per second on a CPU using the RGB and infrared cameras for Full High Definition (Full HD) resolution, faster than other published detectors.**



***Index Terms*—Dual-camera sensors, efficient detector, face detection, high resolution, real-time, low-cost devices.**

I. Introduction

FACE detection is a vision method that can identify and predict the location of human faces in an image. It is an active research field that has continued growing over the past few years. Besides, this method is necessary for various applications and is a supporting process for high-level facial classification systems [1]. In recent years, with the rapid development of intelligent systems, video surveillance was also an outbreak as a solution to prevent criminal acts. Intelligent Surveillance Systems (ISS) play an essential role in these systems. They can automatically analyze the content of video streams from cameras or CCTVs and respond periodically to the abnormal behavior of the monitoring system [2]–[4]. The activity and identity of the person are identified as helpful information. Therefore, this application requires a face detection process at the beginning stage and will be used for the advanced facial analysis process [5]. In general, the video input of this system is high resolution, which demands operation in real-time. Additionally, practical applications encourage this system to work quickly with a minimum delay while maintaining the detector's accuracy [6], [7]. Video surveillance is needed to work all the time. Instead of only utilizing RGB camera sensors, this system also uses infrared to capture objects in the nighttime [8]. Therefore, the face detector should also be robust work in low-illumination challenges.

Recently, deep learning methods for face detection are emerging due to their superior performance [9]–[12]. The CNN method can learn from large-sized datasets and deliver satisfying results in various image conditions [13]–[15].

Manuscript received August 8, 2021; revised October 26, 2021; accepted November 7, 2021. Date of publication November 16, 2021; date of current version December 29, 2021. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Government through Ministry of Science and ICT (MSIT) under Grant 2020R1A2C200897212. The associate editor coordinating the review of this article and approving it for publication was Dr. Xiaojin Zhao. *(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: dputro@mail.ulsan.ac.kr; ndlinh301@mail.ulsan.ac.kr; acejo@ulsan.ac.kr).

Digital Object Identifier 10.1109/JSEN.2021.3128389

TABLE I
PROS AND CONS OF LATEST CPU-BASED FACE DETECTOR

Detectors	Pros	Cons
FCPU [6]	The sequential mini-inception efficiently discriminates the specific features to increase the accuracy.	This detector weakly recognizes small facial features.
RetinaFace-mobile0.25 [10]	A features pyramid network helps the model to obtain various level features in different frequencies.	This model generates a huge computational overhead and operates slowly on a CPU device.
FaceBoxes [17]	Rapidly Digested Convolution Layers (RDCL) and Multiple Scale Convolution Layers (MSCL) extracts features face efficiently.	An anchor densification strategy increases the computation in the detection layer.
FlashNet [18]	It utilizes a light backbone and uses an anchor-free method in prediction layers. It generates a few parameters.	It has a dependency on the TVM optimizer, which increases the data processing speed of the detector model.
DCFPN [19]	A fair L1 loss function regresses box relative center and size in order to locate a small face well.	Densely Connected Convolutional Layers (DCCL) generates more parameter and decrease processing speed of detector.

Li *et al.* [11] proposed a face detector to classify with multi-view, occlusions, and extreme expressions. A bi-channel network and self-learning method are applied to offer better performance in detecting challenging faces. Furthermore, a receptive field task cascaded CNN has been proposed as an accurate face detector [12]. It improves the Inception-V2 module to enhance the feature discriminability and robustness for small targets. The robustness of various methods is summarized in the WIDER face detection benchmark. Some of them achieve high performance [9], [10], [12]. However, the strong performance comes with a drawback in computation cost. It either requires powerful hardware or a lot of processing time. These two factors restrict the applicability of deep learning-based face detectors in various use cases that demand real-time processing or deployment on systems without an expensive device [16].

The CPU face detectors have been developed to enable lower computation costs and work on low-cost devices. The FaceBoxes [17] detector applies Rapidly Digested Convolution Layers (RDCL) and Multiple Scale Convolution Layers (MSCL) to extract features efficiently. In addition, it utilizes an anchor densification strategy to increase the accuracy and recall rate of tiny faces. Other detectors apply anchor-free methods and improve MobileNetV2 to detect faces of various scales [18]. FlashNet overcomes the overhead of computational cost and achieves a small number of parameters. It runs in real-time at 28.30 FPS on a single CPU. However, FaceBoxes and FlashNet are designed to work with small resolution images. These methods were unsuitable for processing higher resolution images required for a surveillance system to identify people's faces. In order to compare the latest CPU-based detector, the summarized pros and cons of each architecture are shown in Table I.

The works in this research aim to develop a high-quality face detection algorithm that works in real-time for high-resolution video. It utilizes a deep learning model that produces high accuracy results due to its capability of learning from a large amount of training data. However, deep learning models often suffer in processing time due to their computation cost [20], [21]. Therefore, high-performance GPUs are required to make the model run smoothly. The usage of an accelerator leads to high hardware costs and makes the system hardware-dependent [22]. This research aims to decrease the computation cost of a deep learning-based face detector to run on a CPU without ignoring the performance. Thus, eliminating the need for expensive GPUs and making the model more hardware independent. Additionally, it also will increase usability when implemented on a low-cost device. The main contributions are as follows:

1) This paper proposes a new framework for a fast and accurate face detector (FAFCPU) that works smoothly on a CPU using the efficient and lightweight CNN architecture. It consists of two main modules, an efficient backbone that is useful to extract distinctive facial features and a four-level detection module to predict variations in facial scales.
2) The proposed model achieves high accuracy and outperforms the other state-of-the-art CPU-based methods in Face Detection Data Sets and Benchmarks (FDDB) [23] and WIDER FACE [24].
3) The proposed FAFCPU detector works using RGB and Infrared cameras in real-time on the CPU. It achieves 27 FPS for high-resolution input video (Full HD) and 122 FPS for VGA input video. These processing times are 50% and 35% faster than the previous state-of-the-art CPU-based detector for full HD and VGA resolution, respectively.

The remainder of the paper is organized as follows: Section II describes the proposed architecture. Section III explains the implementation setup for training and testing of the CNN model. Section IV discusses the experiments conducted to assess the effectiveness and efficiency of the proposed detector. Finally, conclusions and future work are presented in Section V.

II. FACE DETECTOR ARCHITECTURE

The efficient model for real-time facial detection is described in this section. Fig. 1 shows the overall architecture that consists of an efficient backbone, a stem, a transition, an enhancement module, and a four-stage detection layer.

A. Light Backbone Module

Feature extraction plays an essential role for the accuracy of an object detector. The principal of the CNN-based architecture is to extract the pixel information and shrink the

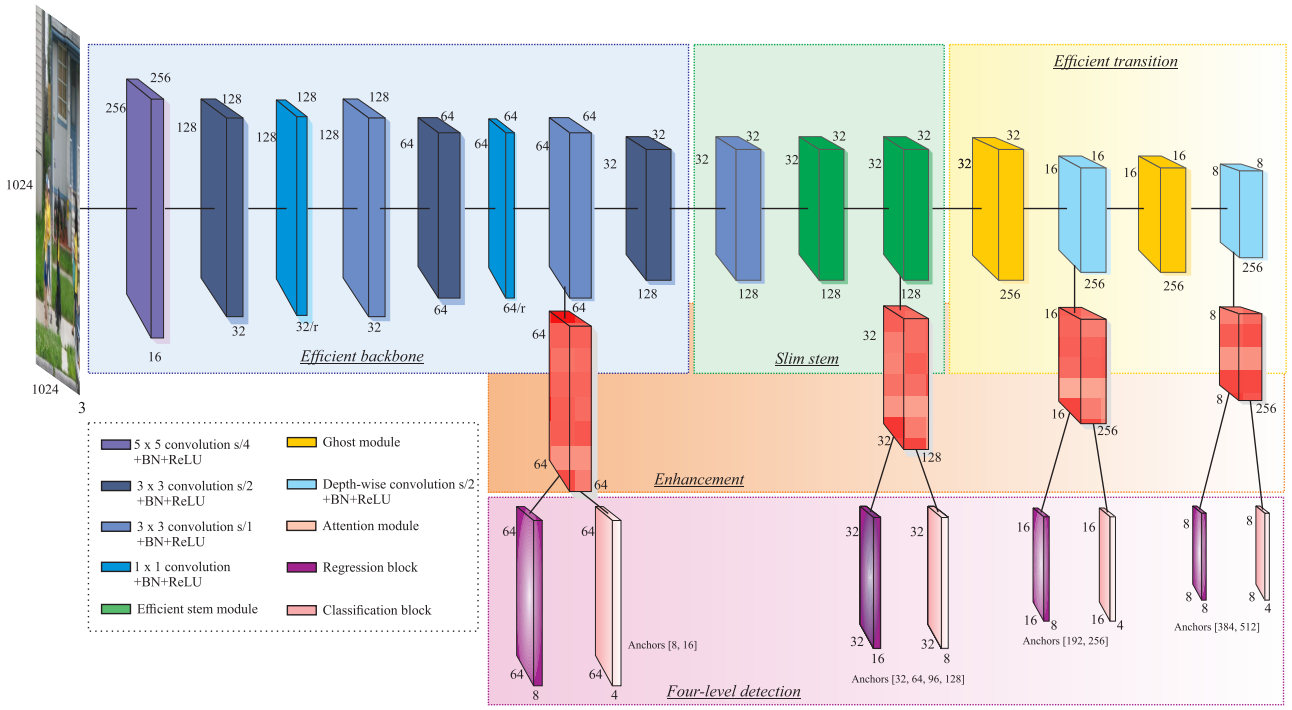


Fig. 1. The FAFCPU architecture contains a light backbone to efficiently discriminate facial features and a four-level detector to predict multiple scales faces. The enhancement module highlights the useful features and reduces trivial features.

feature map at the end of the network [25]. This approach helps to reduce the computation time. On the other hand, it increases the number of channels and the depth to prevent the loss of useful information. As the layers become deep, the computation cost and the number of parameters increase [26]. Therefore, the proposed backbone uses only eight convolution layers to extract the spatial input size rapidly. This module gradually shrinks the feature map to 32 times smaller without ignoring important information from objects. A 5×5 convolution is applied at the beginning stage to reduce the feature map size drastically. Furthermore, the proposed module utilizes the bottleneck convolution series [27] to extract shrunken feature maps efficiently. It uses a smaller number of channels in the 1×1 convolution layer by applying a 0.5 channel ratio of the input features. This technique is used to reduce the number of parameters generated by the next convolution operation. A 3×3 filter is then applied to extract the local features to produce the same size feature map. A shrink block follows these three sequential modules to reduce the size of the feature map. Instead of using a pooling layer, it employs a 3×3 convolutional layer with a stride of two as a local filter. Each convolution layer is followed by Batch normalization and Rectified Linear Units (ReLU) activation for convergence training and to prevent overfitting [28].

B. Efficient Stem Module

The slim stem module is introduced in this work to extract specific facial features comprehensively. This process maintains the size of the feature map, so it sequentially applies the convolution operations. Fig. 2 (a) shows that two 3×3 filters are employed for a stem block by reducing the channel size at the beginning of the layer. It also adopts a bottleneck technique to halve the number of channels from the feature map input that impacted saving trainable parameters. This layer produces medium and high-level features that contain complex facial features. Therefore, a detection layer is installed at the end of this module to predict medium-sized faces. To reduce computations, the proposed model ignores the residual technique [27]. In addition, the use of a lot of stem blocks also affected the increase in the number of parameters. The module efficiency emphasizes compression of computational cost to allow the detector to run fast in real-time without using an additional expensive device.

C. Efficient Transition Module

The transition module utilizes light convolution operations to transform feature maps from medium to high-level prediction layers. It consists of a ghost module [29] and Depth-wise convolution [30], which employs a combination of single channel and simple filter convolution. The ghost module applies a 1×1 filter to the input features ($h \times w \times c$) with the target channel (c^*), as shown in Fig. 2 (b). ReLu and batch normalization are involved after this filter. Furthermore, Depth-wise convolution extracts single channel-based features. The fusion stage is used at the end of the block to combine the extracted features of simple filter and single channel-based images. It produces twice the target channel size ($2c^*$) to enrich the information from the efficiently extracted feature map. Furthermore, a lightweight shrink block is applied to degrade the feature map size between prediction layers. Instead of using pooling to reduce the size, Depth-wise convolution with stride two is more powerful without changing the channel

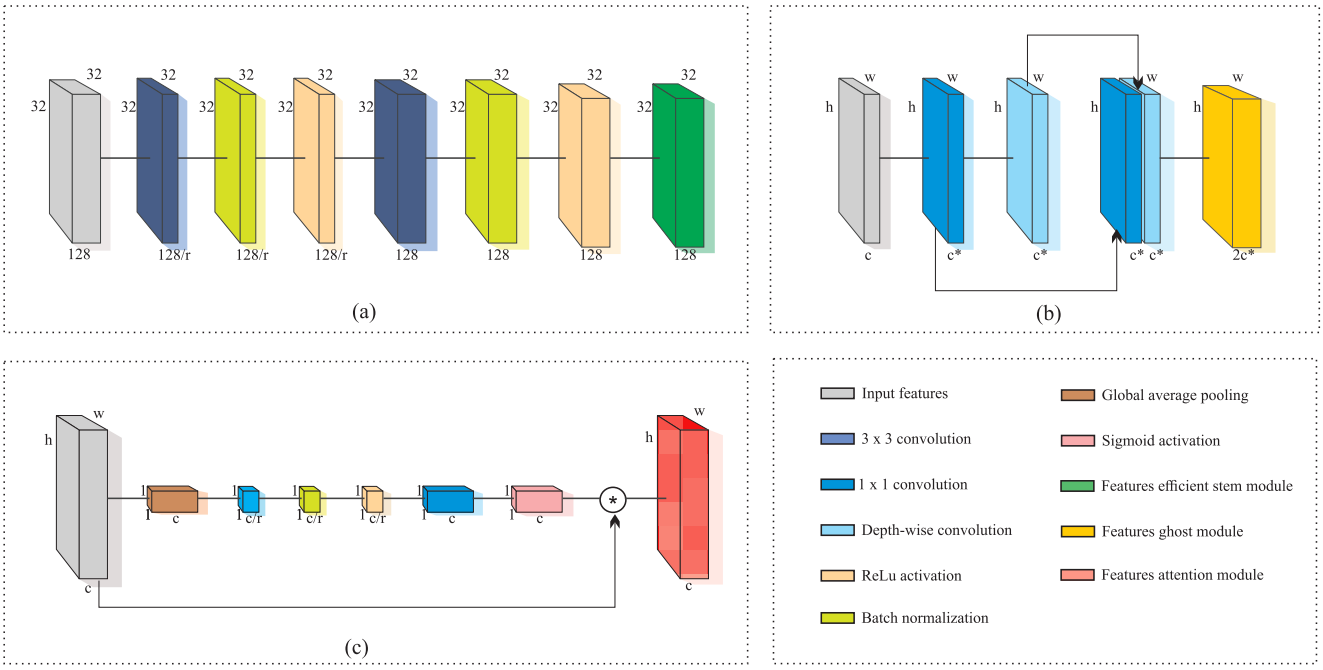


Fig. 2. Proposed efficient stem module using the bottleneck approach (a), ghost module (b), and enhancement module to increase interesting facial features (c).

size. The transition modules efficiently extract and shrink feature maps to assist multi-scale predictor tasks.

D. Four-Level Detection With Anchor Assignment

The FAFCPU uses the pyramidal feature hierarchy to detect faces of various sizes. It avoids extra computations produced by the Feature Pyramid Network (FPN) structure [31]. It eliminates additional operations such as convolution and upsampling connecting between the heads. A head module applies two 3×3 convolutions to generate the regression and classification layers with channel sizes of $coordinates \times anchors$ and $classes \times anchors$, respectively. One stage detector predicts bounding boxes based on dimensional clusters using anchor boxes. It employs various sizes of anchors according to the face scale. Instead of using a single predictor, FAFCPU employs four prediction layers to accommodate multi-scale faces. This structure overcomes the problem of inconsistency between the fixed receptive field and the different facial scales. Additionally, it also assigns different sizes of anchors to each prediction level. The first level is responsible for large faces by applying anchors with 384 and 512 sizes. The second, third, and fourth levels handle medium and small faces by using anchor sizes of [256, 192], [128, 96, 64, 32], and [16, 8], respectively. This strategy focuses on assigning anchors to predict faces based on the scale to increase the effectiveness of the multi-level prediction layer.

E. Enhancement Module

A light extractor feature weakly discriminates against unclear features. Therefore, we introduced an enhancement module to increase the intensity of essential facial information and reduce trivial features. This module globally captures long dependencies to enhance the valuable information and the relationships between facial components. Global average pooling (GAP) is applied to summarize the set of the input features (x_i) for each channel and to convert them into the probability weights, illustrated as:

$$Em_i = x_i \cdot \sigma(W_2 ReLU(W_1 GAP(x_i))). \quad (1)$$

This module is inspired by the SE block [32] but is improved by applying a convolutional squeeze to extract vector information and employing sigmoid (σ) to generate weighted representations. Then, it is used to update the set of pixels from the input features (x_i) as shown in Fig. 2 (c). This module adaptively works with different size features for each prediction level. Therefore, it amplifies the useful information at each prediction layer to improve the detector's performance and minimize prediction errors.

F. Dual Loss Functions

A face detector localizes the vital area of the suspected face by predicting the coordinates and size of the box. This generates a coordinate vector (x, y, w, and h) and classes (face and none). Both offsets are produced by a 3×3 convolution in the prediction layers. The channel size is adapted to the number of anchors. The detector requires a loss function to quantify the inaccuracy of the prediction results. It encourages the performance of updating the weights for each filter. The FAFCPU applies two losses consisting of regression loss to determine the prediction error of the bounding box location and classification loss to calculate the prediction error of the presence of faces. Regression loss employs an L2 function [33] that is used to obtain the error prediction of boxes. It sums up all the squared differences between the ground-truth and the

TABLE II
ABLATIVE STUDY OF LIGHT AND COLOR AUGMENTATION DISTORTIONS

Augmentations	Experiments				
	1	2	3	4	5
Brightness		√	√	√	√
Contrast			√	√	√
Saturation				√	√
Hue					√
TPR (FDDB dataset)	0.972	0.975	0.976	0.976	0.978

predicted scores, described as:

$$L_{reg}(r_i, r_i^*) = \sum_{x,y,w,h} (r_i^* - r_i)^2, \quad (2)$$

where r_i^* and r_i are the ground-truth box and coordinate vectors from the predictor location for each i-th anchor, respectively. This function provides a greater penalty than L1 loss, so the network is pressured to work harder on this mission. Moreover, category loss uses Focal loss [34], defined as:

$$L_{cat}(c_i, c_i^*) = -\alpha(c_i^* - c_i)log(c_i), \quad (3)$$

where c_i is the predicted class, c_i^* is the ground-truth label of 1 and α is the constant parameter of 0.75. This loss function overcomes the class imbalance problem by assigning more weights to misclassified examples. It is distinct from the original version, which ignores the variation of the γ parameter to tune the weight of different samples. Dual loss combines two objective losses and applies parameter balancing on both sides. It helps the network to work fairly on both losses. The dual loss boxes detector is described as:

$$L_D(r_i, c_i) = \frac{1}{N}\sum_i L_{reg}(r_i, r_i^*) + \frac{5}{N}\sum_i L_{cat}(c_i, c_i^*), \quad (4)$$

where N is the denominator in both functions defined as the number of matched default boxes. It is paired with two constants to improve network training.

III. TRAINING AND TESTING CONFIGURATION

This section introduces the training dataset, augmentation, and implementation details to optimize the training process.

A. Training Dataset and Augmentation Data

A learning dataset provides knowledge for detectors to recognize characteristics and models of facial features. Therefore, the complexity of data builds a model to be robust in real-case applications. WIDER FACE is a large dataset containing multiple faces with a high degree of variability in scale, pose, exposure, and occlusion. It consists of 32,203 images, including 12,800 in the training category. An augmentation technique is applied to this dataset to enrich the variety of knowledge. This method also prevents overfitting in the training process. Random cropping, scale transformation, color distortion, and horizontal flipping are used to create varying instances [19]. The color and light distortion use random uniform distributed with the interval of $[-100, 100)$, $[0.5, 1.5)$, $[0, 3)$, and $[-18, 18)$ for brightness, contrast, saturation, and hue, respectively. Based on Table II, the ablation study of augmentation distortions proves that each process provides improved performance of the detector. The brightness augmentation has the most significant impact on detector performance. The last process resizes images to a high resolution of 1024×1024 as the input size of the training mode.

B. Implementation Details

The training process utilizes a mini-batch to divide the dataset into small partitions of 32. The model is trained in end-to-end mode. Random weights are initialized at all filters in the beginning process. Then the backpropagation process works to update these weights. It applies Stochastic Gradient Descent (SGD) [35] to optimize neuron weights. Several parameters are set, including the weight decay of $5 \cdot 10^{-4}$, the momentum of 0.9, and the gradual learning rates. The first 300 epochs are applied at a 10^{-3} learning rate, the next 100 at a 10^{-4} learning rate, then 50 at a 10^{-5} learning rate, and the last 20 at a 10^{-6} learning rate. The evaluation stage requires an anchor matching process by selecting 0.5 IoU (Intersection over Union) [36]. To implement the algorithm into a FAFCPU detector, it uses the PyTorch framework. A GTX1080Ti accelerator is used to speed up the training process. A computer with an Intel Core I5-6600 CPU @ 3.30 GHz and 8 GB RAM is used as the main device for testing. The real-case experiment is conducted in a university environment with different lighting conditions to examine the detector performance in real scenarios. Furthermore, the proposed detector was tested with multi-cameras such as color and infrared cameras, alternately used according to day and nighttime.

IV. EXPERIMENTS AND RESULTS

This section discusses an ablative study to examine the effectiveness of each proposed module, the evaluation on benchmarks, the runtime efficiency to compare the speed with other CPU competitors, and the implementation of the proposed detector in a real-time application on low-cost devices.

A. Model Analysis

The proposed modules are examined one by one in the ablative study section. It uses the same training configuration, except for specified changes to the proposed module. Then, it evaluates the true positive rate of each module at 1,000 false positives on the FDDB dataset. Table III shows the experiments are comprehensively tested by gradually replacing the module and analyzing for accuracy, number of parameters, and speed to observe the strength of each module. Firstly, it removes all balancing parameters from regression and classification loss. It has no impact on parameters and speed. However, it only reduces performance by 0.006. Secondly, it ignores the enhancement modules that are applied to each head of a detection layer. It reduces AP and parameters by 0.004 and 19K, respectively. Besides, it increases the speed by 1 FPS. Thirdly, each ghost module is replaced with a 1×1 convolution, adding 47K parameters but only increases AP by 0.002. Fourthly, it does not use four detection layers.

TABLE III

ABLATIVE RESULTS OF THE PROPOSED MODULES. THE EVALUATION CONSISTS OF TRUE POSITIVE RATE (TPR) ON THE FDDB DATASET, THE NUMBER OF PARAMETERS, AND THE SPEED OF MODEL ON A CPU

Exp	Proposed modules						Evaluations		
	Balancing loss	Enhancement	Eff. Transition	Four-level det	Eff. Stem	Backbone	TPR	Parameters	FPS (Full HD)
1	√	√	√	√	√	√	0.978	735,556	27.23
2		√	√	√	√	√	0.972	735,556	27.23
3			√	√	√	√	0.968	716,524	28.24
4				√	√	√	0.970	763,116	28.87
5					√	√	0.923	728,520	31.35
6						√	0.899	285,128	39.36

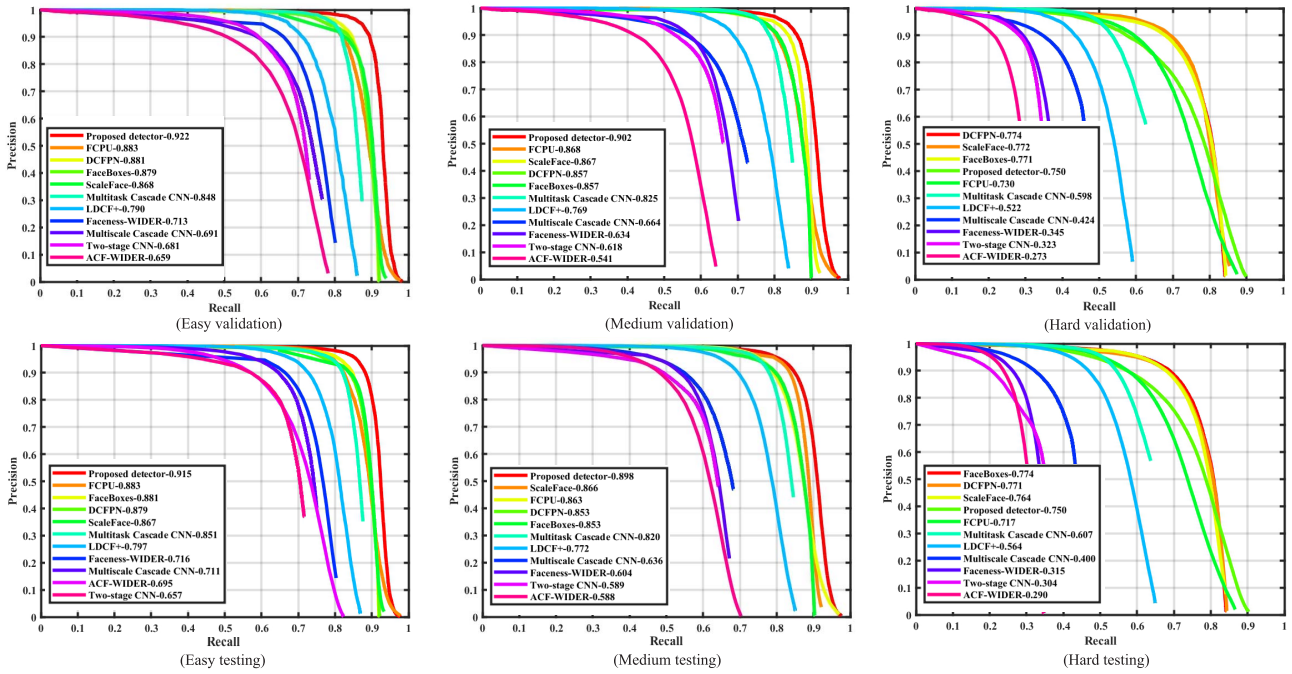


Fig. 3. Evaluation on the WIDER FACE validation and testing sets.

It employs a prediction layer by applying all anchors in a layer. Even though this experiment increased the speed by 2.48, it significantly decreased AP and parameters by 0.047 and 35K, respectively. Finally, the efficient stem block significantly increases the performance by 2.4%, adding to the parameters by 443K. On the other hand, it only reduces the detector speed by 8 FPS.

B. Benchmark Evaluation

The evaluation of the proposed detector is presented on the benchmarking datasets FDDB and WIDER FACE. It also compares the performance with those of other detectors.

1) WIDER FACE Dataset: This dataset is a large face benchmark that contains many variations in challenges, such as scales, poses, expressions, occlusions, and lighting. It is divided into training (40%), validation (10%) and testing (50%) sets. Each validation and testing set provides easy, medium, and hard categories. Fig. 3 shows that the FAFCPU obtains performances of 0.922 (easy), 0.902 (medium), and 0.75 (hard) on the validation sets, while the testing sets are 0.915 (easy), 0.898 (medium), and 0.75 (hard). It outperforms the FCPU [6], FaceBoxes [17], and DCFPN [19] detectors on the easy and medium validation and testing sets. However, FAFCPU is inferior to FaceBoxes and DCFPN for the hard category. Fig. 5 (a) shows that the FAFCPU detector can

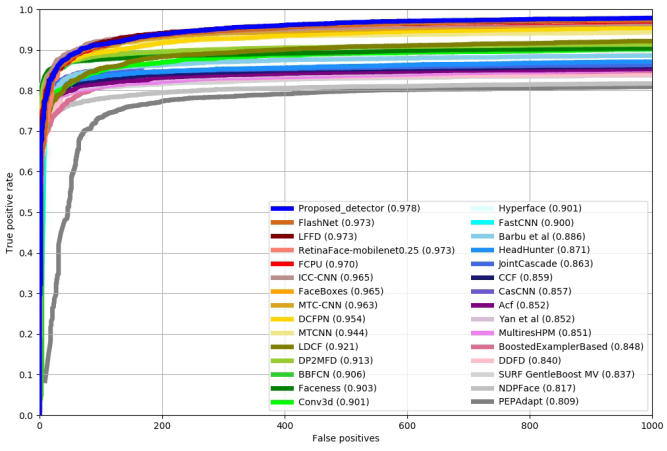


Fig. 4. Evaluation results using discrete ROC (Receiver Operating Characteristics) curves with 1,000 false positives on the FDDB dataset.

Fig. 5. Qualitative results on the WIDER FACE dataset (a), FDDB dataset (b), FDDB dataset with distortions (c), real-time color video (d), and real-time night vision video on Full HD resolution (e).

detect multiple small faces. The proposed detector lacks the ability to predict tiny faces. The shallow backbone is too weak to distinguish small facial features, so that fourth level detection often mistakenly predicts faces with these specific features.

2) FDDB Dataset: This dataset has 2,845 images containing 5,171 faces collected from Yahoo websites. The images consist of various resolution, e.g. 363 × 450 and 229 × 410. It includes faces of famous people with position, lighting, and background challenges. Fig. 5 (b) shows that the FAFCPU can detect the faces of these challenges. This detector achieves a True Positive Rate (TPR) performance of 0.978, as represented in Fig. 4. These results show that it is superior to the state-of-the-art CPU detectors. The evaluation stage requires a conversion process to obtain a rectangle box. It applies discrete criteria by comparing the intersection between the prediction and ground truth. This means the actual score will be one when the IoU is higher than 0.5 and 0 otherwise. The

TABLE IV
EVALUATION OF DETECTOR IN DIFFERENT LIGHT AND COLOR DISTORTIONS ON FDDB DATASET

Experiments	Proposed		FCPU [6]	
	TPR	Time (ms)	TPR	Time (ms)
Without distortion	0.978	0.0075	0.970	0.0096
Bright	0.974	0.0076	0.964	0.0095
Bright+Contrast	0.972	0.0076	0.962	0.0096
Bright+Contrast+Saturation	0.970	0.0076	0.955	0.0095
Bright+Contrast+Saturation+Hue	0.968	0.0075	0.955	0.0096

proposed detector is also examined at different light and color distortion, as shown in Table IV. Entire images in the FDDB dataset are sequentially applied to scenario distortions such as brightness, contrast, saturation, and hue. As a result, the FAFCPU detector only degrades by 1% when applied to all scenarios. The visualization results show proposed detector successfully detects the face in dark and light conditions, as shown in Fig. 5 (c). In contrast, the latest competitor is more affected by the distortions process. In addition, these experiments show that both detectors have a similar average time in the inference phase. The various distortion images have no impact on the processing time of a CNN-based detector. The speed of the spatially based convolution operation is only affected by the dimension of feature map input and kernel size. Based on the evaluation result, the proposed detector has a robust feature extractor that efficiently discriminates the facial features. The combination module effectively predicts multi-scaled faces so that the architecture can be adopted by CNN-based face classifiers such as facial recognition, emotion, gender, age, and others.

C. Runtime Efficiency on Different Video Resolutions

A CNN-based detector generally produces high computation because it employs many operations to generate the essential features. It requires an expensive device to run in real-time, so it can be accelerated using a GPU. However, practical applications demand a vision method to operate on CPUs with low computational complexity and lightweight parameters. The FAFCPU generates 735,556 parameters with 300 MFLOPS. These results indicate that the proposed detector is slighter than the standard CNN-based architecture. In the testing stage, the experiments are conducted on a computer with an Intel Core I5-6600 CPU @ 3.30 GHz and 8 GB RAM. It also compares its speed with the competitors on different video resolutions, such as VGA (Video Graphics Array), HD (High Definition), and Full HD (Full High Definition). The final bounding boxes are produced by applying a Non-Maximum Suppression (NMS) of 0.3 to select positive anchor boxes and a confidence threshold of 0.05.

As a result, the FAFCPU outperforms the speed of the FCPU as the latest competitor. Fig. 6 shows a significant difference in VGA-resolution of 33 FPS. In addition, the proposed detector also runs faster than the FlashNet detector [18], which differs by 0.005 AP on the FDDB dataset. Even the FAFCPU achieved 27.23 FPS for Full HD resolution video, which stands out from other detectors that fail to achieve real-time speeds.

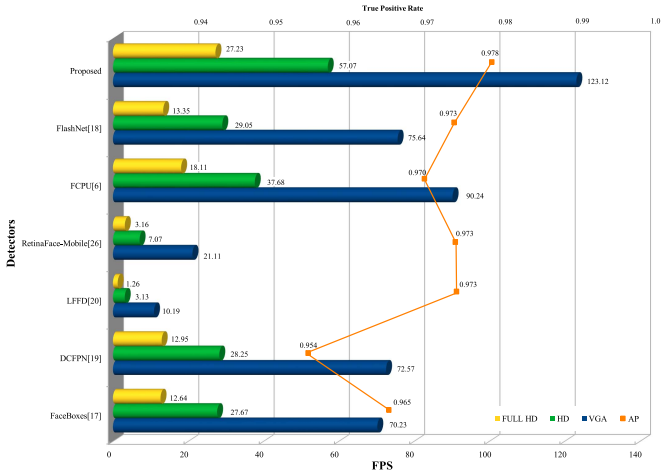


Fig. 6. Comparison of detector data processing speeds at different video input sizes.

TABLE V
THE FAFCPU SPEED USES MULTIPLE CAMERAS IN DIFFERENT RESOLUTION VIDEOS

Num. of cameras	VGA		HD		FULLHD	
	Time (ms)	FPS	Time (ms)	FPS	Time (ms)	FPS
1	0.0081	123.12	0.0175	57.07	0.0367	27.23
2	0.0163	61.51	0.0352	28.4	0.0736	13.58
3	0.0242	41.25	0.0527	18.98	0.111	9.01
4	0.0325	30.77	0.0704	14.21	0.1475	6.78

Fig. 5 (d) shows the visualization results of the proposed detector working for Full HD resolution video that can detect multi-profile faces. In addition, FAFCPU is also robust in recognizing facial features on low-illuminance conditions with infrared-based cameras, as shown in Fig. 5 (e). Although some faces are covered with masks, this challenge does not obstruct the detector from finding the face location. Video surveillance generally uses multiple cameras to monitor objects and people. Table V shows that the proposed detector is tested for data processing speed in a single system that uses a different number of cameras. Since a system handles more than one camera, the detector speed is also divided according to the number of cameras used. However, it can run in real-time on VGA and HD resolutions. In contrast, it runs quite slowly when using more than two cameras at FULL HD resolution. The proposed detector comprehensively learns specific facial features from complex data sets. It robustly discriminates against the human face for video surveillance even at various exposure distortions. In addition, the learning-based model emphasizes efficiency to be able to operate in real-time on high-resolution input.

D. Real-Time Application on Low-Cost Devices

Practical applications encourage a vision-based detector to work in real-time. In addition, its implementation on a low-cost device is more valuable because these devices are more widely used for current technologies. PC, Lattepanda, and Notebook use CISC processors (Complex Instruction Set Computer), while Raspberry Pi represents RISC processors

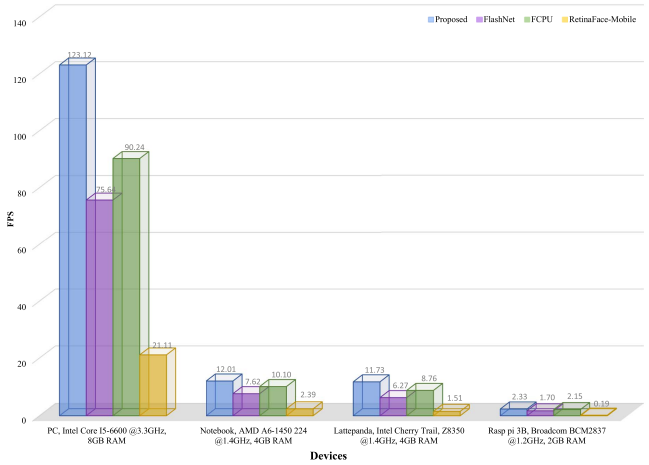


Fig. 7. Comparison of detector speeds on low-cost devices.

(Reduced Instruction Set Computer). CPU detector's speeds were examined for efficiency with several inexpensive devices on VGA-resolution. We take input from a webcam to analyze the real-time speed of each detector at 1,000 frames. Fig. 7 shows that the FAFCPU is superior when operated in real-time applications. Each device produces a different speed depending on the processor size. The proposed detector runs slowly on raspberry pi, but it works smoothly on Notebook and Lattepanda devices. In contrast, other detectors work slowly on all three devices. Therefore, the FAFCPU is more efficient than CPU-based face detectors on low-cost devices. The proposed architecture avoids redundant operations, thus producing a low amount of computations. In addition, the proposed modules maintain the performance of the detector to recognize multiple-scale faces with high accuracy.

V. Conclusion

This paper presents a dual camera-based fast face detector that works in real-time on a CPU using the CNN method. A high-performance detector contains several light modules that generate fewer parameters than the general CNN method. It consists of two main modules, an efficient backbone, and a four-level detection module. Efficient transition and stem modules are applied to distinguish distinctive features rapidly. It also produces less computation cost, which allows the detector to work quickly. In addition, several training parameters and dual loss functions improve the training performance of the model. As a result, the FAFCPU achieves state-of-the-art performance on the face detection benchmarks compared with CPU-based detectors. This detector is fastest than other competitors and achieves real-time speeds at a Full HD resolution of 27 FPS on a CPU. The proposed detector can detect multiple faces in different light distortions when implemented in real scenarios. In future work, a super-resolution method can be explored to increase the accuracy of tiny face detection. Additionally, advanced applications in metropolitan areas will be developed to enhance the capabilities and capacities of face detectors.

References

[1] C.-H. Lin, Z.-H. Wang, and G.-J. Jong, "A de-identification face recognition using extracted thermal features based on deep learning," *IEEE Sensors J.*, vol. 20, no. 16, pp. 9510–9517, 2020.

[2] A. Filonenko *et al.*, "Unattended object identification for intelligent surveillance systems using sequence of dual background difference," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2247–2255, Dec. 2016.

[3] A. Shahbaz and K.-H. Jo, "Deep atrous spatial features-based supervised foreground detection algorithm for industrial surveillance systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4818–4826, Jul. 2021.

[4] C. Vishnu, R. Datla, D. Roy, S. Babu, and C. K. Mohan, "Human fall detection in surveillance videos using fall motion vector modeling," *IEEE Sensors J.*, vol. 21, no. 15, pp. 17162–17170, Aug. 2021.

[5] G.-S. Hsu and T.-Y. Chu, "A framework for making face detection benchmark databases," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 230–241, Feb. 2014.

[6] 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.

[7] Q. Hu, X. Tang, and W. Tang, "A real-time patient-specific sleeping posture recognition system using pressure sensitive conductive sheet and transfer learning," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6869–6879, Mar. 2021.

[8] A. Shahbaz and K.-H. Jo, "Dual camera-based supervised foreground detection for low-end video surveillance systems," *IEEE Sensors J.*, vol. 21, no. 7, pp. 9359–9366, Apr. 2021.

[9] S. Zhang, C. Chi, Z. Lei, and S. Z. Li, "RefineFace: Refinement neural network for high performance face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4008–4020, Nov. 2021.

[10] T. M. Hoang, G. P. Nam, J. Cho, and I.-J. Kim, "Deface: Deep efficient face network for small scale variations," *IEEE Access*, vol. 8, pp. 142423–142433, 2020.

[11] J. Li, L. Liu, J. Li, J. Feng, S. Yan, and T. Sim, "Toward a comprehensive face detector in the wild," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 104–114, Jan. 2019.

[12] X. Li, Z. Yang, and H. Wu, "Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks," *IEEE Access*, vol. 8, pp. 174922–174930, 2020.

[13] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "TSE-CNN: A two-stage end-to-end CNN for human activity recognition," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 1, pp. 292–299, Jan. 2020.

[14] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de-Gabriel, "CNN-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors J.*, vol. 19, no. 16, pp. 6872–6882, Aug. 2019.

[15] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors J.*, vol. 19, no. 7, pp. 7598–7604, Sep. 2019.

[16] F. T. El-Hassan, "Experimenting with sensors of a low-cost prototype of an autonomous vehicle," *IEEE Sensors J.*, vol. 20, no. 21, pp. 13131–13138, Jun. 2020.

[17] S. Zhang, X. Wang, Z. Lei, and S. Z. Li, "Faceboxes: A CPU real-time and accurate unconstrained face detector," *Neurocomputing*, vol. 364, pp. 297–309, Oct. 2019.

[18] Y. Ge, Q. Wang, B. Sheng, and W. Yang, "FlashNet: A real-time anchor-free face detector," in *Proc. 35th Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Oct. 2020, pp. 441–446.

[19] S. Zhang, X. Zhu, Z. Lei, X. Wang, H. Shi, and S. Z. Li, "Detecting face with densely connected face proposal network," *Neurocomputing*, vol. 284, pp. 119–127, Apr. 2018.

[20] Y. He, D. Xu, L. Wu, M. Jian, S. Xiang, and C. Pan, "LFFD: A light and fast face detector for edge devices," 2019, *arXiv:1904.10633*.

[21] N. Zhou, R. Liang, and W. Shi, "A lightweight convolutional neural network for real-time facial expression detection," *IEEE Access*, vol. 9, pp. 5573–5584, 2021.

[22] M. D. Putro, D.-L. Nguyen, and K.-H. Jo, "Lightweight convolutional neural network for real-time face detector on CPU supporting interaction of service robot," in *Proc. 13rd Int. Conf. Hum. Syst. Interact. (HSI)*, Jun. 2020, pp. 94–99.

[23] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. UM-CS-2010-009, 2010. [Online]. Available: http://vis-www.cs.umass.edu/fddb/index.html

[24] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5525–5533. [Online]. Available: http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/index.html
[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[26] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5203–5212.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.
[29] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1580–1589.
[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[31] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Dec. 2018, pp. 7132–7141.
[33] K. Janocha and W. Marian Czarnecki, "On loss functions for deep neural networks in classification," 2017, *arXiv:1702.05659*.
[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[35] S. Al-Momani, H. Mir, H. Al-Nashash, and M. Al-Kaylani, "Brain source localization using stochastic gradient descent," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8375–8383, Mar. 2021.
[36] A. Nandy, "A densenet based robust face detection framework," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1840–1847.

Muhamad Dwisnanto Putro (Graduate Student 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 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.

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

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. He is currently serving as the Faculty Dean for the School of Electrical Engineering, University of Ulsan. His research interests include computer vision, robotics, autonomous vehicles, and ambient intelligence. Dr. Jo has served as the Director or an AdCom Member for 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 in 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*.