Vision-Based People Counter Using CNN-Based Event Classification

Sung In Cho, *Member, IEEE*

***Abstract*— This article proposes a convolutional neural network (CNN)-based people counter that classifies a given frame cube to a specific event that indicates people entering or exiting a target area to measure instantaneous people count. For the training of the proposed CNN, a training input frame cube and its corresponding class label that represents a specific event are generated using the proposed counting rules. For mitigating the overfitting problem that may occur in the training of the proposed CNN, data augmentation, and postclass correction using foreground distribution with event probabilities are applied. The experimental results indicate that the proposed method improved the F1 score and accuracy for the cumulative people counting results by up to 9.0% and 14.8%, respectively, compared with those of the benchmark methods, even though it calculated the cumulative count by summing instantaneous people counts, while the benchmark methods were optimized for the calculation of the cumulative count.**

***Index Terms*— Convolutional neural network (CNN), data augmentation (DA), event classification, people counting.**

I. Introduction

THE vision-based measurements have been widely used in various applications in recent years [1]. Among these, a vision-based people counter that derives the number of people entering and leaving a specific area for a given image is used widely in various applications, such as video surveillance, urban planning, resource management, and customer profiling. In particular, the information of people count is highly useful for retail shops or restaurants because it can be used for the analysis of customer visit patterns, which could enable efficient management. Further, if the information on people count in various retail shops in a specific area is available, it can be used purposefully for regional commercial analysis.

Thus, various vision-based people counters have been developed. Most of the people counters use a line of interest (LOI)- or region of interest (ROI)-based approach to measure the number of people moving in a target area. These people-counting methods can be categorized as detection with tracking-based methods [2]–[5] and segmentation with regression-based methods [6]–[12]. The tracking-based methods detect people's locations and track them to extract the people count. In [2], to detect individual entities, people's locations are extracted and tracked using unsupervised Bayesian clustering. In [3], heads are detected using the 2-D correlation between a bank of annular patterns and the extracted foreground (FG) regions. Subsequently, a Kalman filter is used for tracking detected heads, and the people count estimated using an LOI-based approach. However, these detections with tracking-based methods are highly vulnerable to changes in environmental illumination and occlusion in the environments, where several people overlap each other in the image or appear and disappear. Thus, the accuracy of people counting can deteriorate significantly in such cases.

Methods based on segmentation with regression generally use FG and motion information for crowd segmentation and extracting people counts. Flow mosaicking (FM) [6] is the most popular motion-based people counter. In this method, the motion vectors (MVs) on a selected LOI are extracted using an optical-flow-based approach. Subsequently, a temporal slice image (TSI) is generated by stacking LOI pixels over time. The moving segment is stacked to produce a blob, and the size of each segment stacking is determined by its velocity that can be calculated by the magnitude of the MV. The final cumulative people count of a given video is calculated by a regression function with the volume of the extracted blob. In [7], a crowd is segmented into subcomponents of homogeneous motions using a combination of dynamic-texture motion models. Subsequently, a low-level feature is extracted from each segment. The number of people is calculated from the Bayesian Poisson regression (BPR) using the extracted features. In [8], a TSI is generated by stacking LOIs over time as in FM [6], but a fixed-line width is used for stacking, whereas FM uses the velocity of a moving segment as the size of the line stacking. Subsequently, people count is estimated in a set of overlapping sliding windows called the temporal ROI (TROI) on the TSI using a regression function. People count in each TROI is estimated using BPR with a local histogram of oriented gradients and various features. In [9], the method of [8] is improved using a multiple-window-length TROI and an outlier-robust objective function. These segmentations with regression-based methods estimate the number of people in the existing data sets successfully [7], [10], [13] for the evaluation of various people counters, which were captured in an environment, where the distance between the people and camera is sufficiently large, as shown in Fig. 1. However, these existing methods are disadvantageous

Manuscript received July 1, 2019; revised October 15, 2019; accepted December 4, 2019. Date of publication December 16, 2019; date of current version June 24, 2020. This work was supported in part by the Dongguk University Research Fund of 2019 and in part by the Korea Government through the National Research Foundation of Korea (NRF) [Ministry of Science, Information and Communication Technology (ICT) and Future Planning (MSIP)] under Grant 2017R1C1B5075091. The Associate Editor coordinating the review process was Amitava Chatterjee.

The author is with the Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea (e-mail: csi2267@dongguk.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2019.2959853

(a) (b) (c)

Fig. 1. Examples of conventional data sets. (a) Frame from the University of California at San Diego (UCSD) data set [7]. (b) Frame from the Grand Central data set [10]. (c) Frame from the Mall data set [13].

for indoor environments, such as a small retail shop, where the camera and people are close to each other, and the motions of a person entering or leaving are generally much larger than those in the existing data sets, as shown in Fig. 2. In addition, in these environments, luminance variation and occlusion occur frequently. Consequently, the accuracy of the existing people counters can be degraded significantly.

To alleviate these problems, in our previous works [14], the flow volume analysis-based people counter (FAPC) that can consider the practical environments of the people counting problem was proposed. In this method, the multiple-touching section-based FG analysis and the maximum *a posteriori* probability-based dilated motion estimation were proposed to improve the accuracy of people counting in practical retail shops. While the FAPC could provide better accuracy of people counting compared with the existing methods [2]–[12], the performance can still be improved. In addition, the FAPC could not provide an accurate instantaneous people count because it focuses on deriving the accumulated count for each input sequence. Specifically, the FAPC utilized a multivariate linear regression function on an accumulated count for each test set; thus, it generally provided the lower accuracy of an instantaneous count than the accuracy of a cumulative count.

In addition to these model-based people counters, deep learning-based people counters had recently been proposed to take advantage of the outstanding feature extraction capability of deep learning [12], [15]–[18]. Most deep learning-based people counters [12], [15], [16] aim to estimate the number of people in a large crowd of a given image. For LOI-based people counting, which estimates the number of people entering and leaving an LOI, [17] and [18] were proposed. In [17], TSIs were generated by stacking LOI pixels in input frames and optical flow maps. Then, three regression convolutional neural network (CNN) modules that utilize the TSIs, as an input, were used to estimate the number of people entering and leaving. In [18], crowd density and crowd velocity were estimated by using their modules for the LOI-based people counting. These CNN-based people counters provided excellent counting accuracy; however, the method of [17] may be less accurate in indoor environments, where the frame rate of the input is extremely low while the method of [18] requires large training sets with manually labeled trajectories.

Herein, a CNN-based people counter that can provide the outstanding accuracy of instantaneous people count compared with the existing people counters under the indoor people counting environment is proposed. The proposed method is intended for deriving people count by receiving the input frames captured from a large number of retail shops in a small specific area; thus, it can be used for regional commercial analysis. Considering the input frame acquisition and transmission environments of a practical retail shop, it is assumed that a camera is installed on the ceiling of a doorway (a narrow entrance/exit area) over the heads of visitors and that the frame rate of the input images is extremely low (less than five frames per second), as in the previous study [14]. Thus, the input images would generally include a smaller field of view and larger people motions than the existing data sets, as shown in Figs. 1 and 2.

The primary contributions of this article are summarized as follows.

1) In the proposed method, the problem of people counting is converted into an event classification problem of a given input frame. Subsequently, a new CNN-based people counter that can provide accurate instantaneous people count by classifying a given input frame into a specific event indicating the entrance and exit situation of people is proposed. In addition, data augmentation (DA) is used for alleviating the overfitting problem of the proposed CNN.
2) To improve the accuracy of people count derived from the proposed CNN, a postclass correction (PCC) that incorporates the event occurrence probabilities obtained from the proposed CNN and FG distribution is proposed.

II. Proposed Method

The proposed method uses the CNN architecture shown in Fig. 3 to classify the entering or exiting of people to derive the people count of a target region. First, the processes of training data generation and CNN training are explained. Next, the architecture of the proposed CNN is explained.

A. CNN-Based Event Classification

1) Training Data Generation and Counting Rule: For the training of the proposed CNN shown in Fig. 3, the images (shown in Fig. 2) obtained from the camera installed on the door is used as the input. Specifically, images acquired from microcomputers, such as the Raspberry Pi and Pi camera, were used as in [14] in consideration of image acquisition and wireless transmission at a retail shop. As mentioned in [14], a Raspberry Pi with Pi camera is inexpensive, easy to install at retail shops, and can easily deliver captured images to servers. The RGB training image of resolution 640 pixels $\times$ 480 pixels is converted to a grayscale image by averaging the color planes. Subsequently, the converted grayscale image is downsampled using bicubic interpolation to 1/20 of the size (32 pixels $\times$ 24 pixels) before using it as a network input. The principle of downsampling is to convert the input image to the smallest resolution at which the movement of the person can be identified by an observer. It was considered that if an observer could identify the movement of a person in a given image, CNN would also be capable of the same. This downsampling not only results in reduced computational complexity but also mitigates the overfitting problem of the proposed CNN, which can degrade the counting performance.

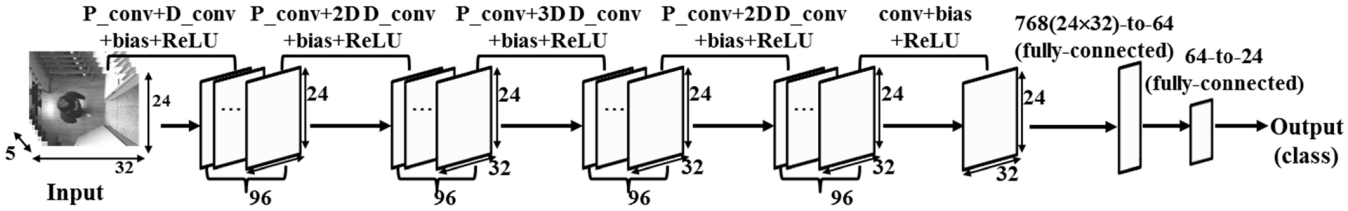

Fig. 2. Example data sets for a practical environment for estimating the number of people entering and leaving a retail shop.

Fig. 3. Proposed CNN-based people counter.

TABLE I
SPECIFICATIONS OF TRAINING DATA PAIR

Data	Dimension	Details
Input frames	32×24×5	[t-2th, t-1th, t^{th}, t+1th, t+2th] frames
Ground-truth (output)	24×1 (8×1 for each input frame)	Output for [t-1th, t^{th}, t+1th] frames - [1:8] for t-1th frames - [9:15] for t^{th} frames - [16:24] for t+1th frames

TABLE II
OUTPUT CLASSES (EIGHT CLASSES)

Event (Class number)	Counting number	Class (Class number)	Counting number
No movement (1)	+0	Left in (2, 3, 4)	+1 or +2 or +3
Right out (5,6,7)	+1 or +2 or +3	Left in and right out (8, Fig. 4 (e))	+1 on each side

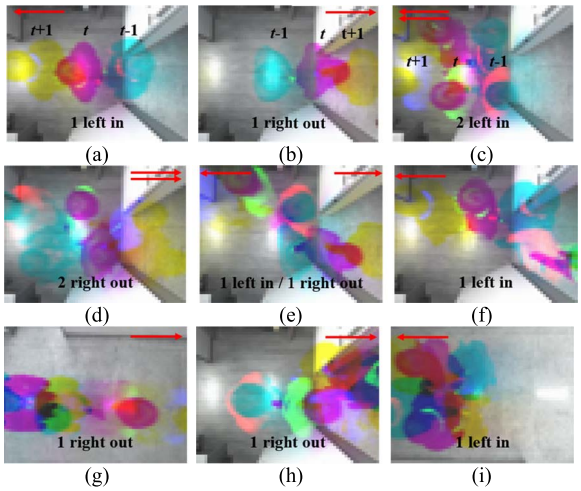

Fig. 4. Example of input frames at the time of counting the number of entering or exiting people (counting condition and counting number). (a) Left in +1. (b) Right out +1. (c) Left in +2. (d) Right out +2. (e) Left in +1 and right out +1. (f) Left in +1. (g) Right out +1. (h) Right out +1. (i) Left in +1.

In addition, multiple input frames are used to construct a framed cube so that the proposed CNN can estimate the direction of FG movement. Specifically, five [($t-$ 2)th – ($t+$ 2)th)] frames are used to construct a frame by considering the field of view and frame rate of input frames. As shown in Fig. 3 and Table I, a framed cube consisting of five consecutive downscaled frames (32 pixels × 24 pixels × 5 pixels) is used as an input to the proposed CNN. For this input frame cube, the proposed CNN provides a classification result that represents the entrance and exit situations of a target area. The output of the proposed CNN is classified into eight cases, as shown in Table II. One hot encoding was used for the output representation; thus, the result of the proposed CNN is denoted by an 8 × 1 output vector. Each element of the output vector represents the probabilities of eight types of classes. For a current input frame cube consisting of the ($t-$ 2)th – ($t+$ 2)th frames, the proposed CNN provides three consecutive output classes that describe the entrance and exit situations of the ($t-$ 1)th – ($t+$ 1)th input frames. Hence, the size of the final output for a given input frame cube is 24 × 1, as shown in Table I.

Fig. 4 shows the examples of frames consisting of the three consecutive frames [($t-$ 1)th – ($t+$ 1th)] for determining the output class (label) for the current time t. In this example, the R, G, and B color planes denote the ($t-$ 1)th, tth, and ($t+$ 1)th input frames, respectively. As shown in Fig. 4, the moment when a part of the person in the ($t+$ 1)th frame leaves, the camera capturing area is defined as the counting time. In other words, counting is performed based on people exiting the ($t+$ 1)th frame and not in the other cases. Fig. 4(a) and (b) shows the simple counting cases in which a person moves in one direction. Specifically, Fig. 4(a) shows the moment when a part of the person in the ($t+$ 1)th frame, represented in yellow, is leaving to the left. Meanwhile, Fig. 4(b) shows the moment when a part of the person is leaving to the right. When multiple people are exiting, as shown in Fig. 4(c) and (d), the moment when some of the members in the $t+$ 1th frame leave the capturing area is defined as the counting time. Fig. 4(e) demonstrates that the people's entries and exits occurred simultaneously. For this case, the incoming (N^{IN}) and outgoing (N^{OUT}) counting numbers increase simultaneously. Fig. 4(f)–(i) shows the complex situations in which additional people movements occur while people are exiting. As in the previous case, only the number of people in the (t + 1)th frame leaving the capturing area is considered for people counting.

Because three consecutive output classes are extracted for each input frame cube, the stride size of the input frame cube

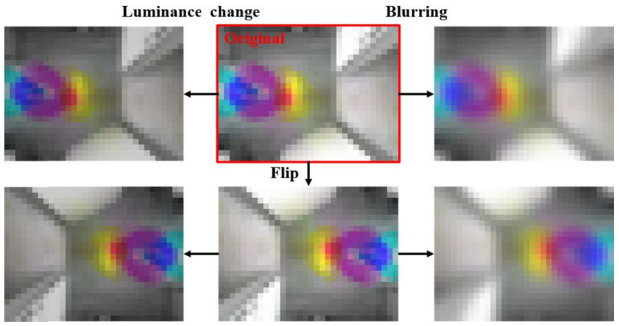

Fig. 5. Augmentation of training data.

TABLE III
STRUCTURE OF THE PROPOSED NETWORK

Layer	Operations	Dimension [S_R, S_C, F_{D1}, F_{D2}]
1st layer	P_conv+D_conv+bias+ReLU	P_conv: 1×1×5×96, D_conv: 3×3×96×1
2nd layer	P_conv+2 dilated D_conv+ ReLU	P_conv: 1×1×96×96, D_conv:3×3×96×1
3rd layer	P_conv+3 dilated D_conv+ ReLU	
4th layer	P_conv+2 dilated D_conv+ ReLU	
5th layer	Conv+bias+ ReLU	Conv: 3×3×96×1
6th layer	96 dense (fully connected)	768×64
7th layer	24 dense (fully connected)	64×24
Total number of weights		83136

generation was set to three. For the training of the proposed CNN, 5491 input frames were used, and the labeling of each input frame cube was performed manually using the counting rules described above.

2) Augmentation of Training Data: Although the training data pair was generated using 5491 frames, the generated training data cannot reflect all possible counting situations, which can cause an overfitting problem of the proposed CNN. Therefore, the training data pairs were augmented to mitigate the overfitting problem, as shown in Fig. 5. First, through image flipping, the object positions in the background (BG) and the movement direction of a person were diversified. Next, the luminance of the original and flipped images was changed by multiplying the original and flipped images by an arbitrary value between 0.8 and 1.2 to consider various light conditions. Finally, blurring operation using a 3 × 3 mean filter was applied on the original and flipped images to consider the variations in light spread condition and image sharpness. By this augmented training data, 5491 × 6 training data pairs could be used for raining the proposed CNN. The improvements in counting accuracy by DA will be analyzed in Section III.

3) Architecture of the Proposed CNN: The structure of the proposed CNN is demonstrated in Fig. 3 and Table III. The proposed CNN performs a convolution operation from the first to the fifth layer to generate feature channels having the same spatial resolution as the input frame. Specifically, the depthwise separable convolution (DSC) [19] consisting of the pointwise convolution (P_conv) and the depthwise (D_conv) convolution was used to effectively reduce the number of weights for the convolution operation while maintaining the network performance. After DSC, a bias value is added, and the rectified linear unit (ReLU) [20] is used as an activation function as follows:

$$\hat{\mathbf{Y}}_{\mathbf{L}} = \begin{cases} \mathbf{W_L} * \hat{\mathbf{Y}}_4 + \mathbf{bias}_5 & \text{if L} = 5 \\ \max\left(\mathbf{W_L^D} * \left(\mathbf{W_L^P} * \mathbf{X^N}\right) + \mathbf{bias_L}, 0\right) & \text{otherwise} \end{cases} \quad (1)$$

where $\mathbf{W_L}$, $\mathbf{W_L^D}$, and $\mathbf{W_L^P}$ denote the weights for the Lth general convolution, D_conv, and P_conv, respectively; $\mathbf{bias_L}$ is the bias vector for the Lth layer; $\mathbf{X_N}$ is a 32 × 24 × 5 input frame cube of the proposed CNN; $\hat{\mathbf{Y}}_{\mathbf{L}}$ is the output layer of the Lth convolution operation; $*$ denotes the convolution operator. As shown in Table III and Fig. 3, when D_conv is applied, dilated convolution [21] is applied to effectively increase the receptive field of each convolution. The dilation size is increased by a half of all the convolution layers and subsequently decreased in the remaining convolution layers. After the generation of feature channels by convolution, a fully connected layer is produced to extract a 64 × 1 feature vector as a result of the sixth layer from the 768 × 1 input vector that is reshaped from the fifth convolution layer. Finally, the output of the proposed CNN is extracted using the fully connected layer that converts a 64 × 1 feature vector to a 24 × 1 output vector as follows:

$$\hat{Y}_L^j = \sum_i^N w_i \cdot Y_{L-1}^i \quad (2)$$

where i and j are indexes for the input and output vectors, respectively, and w denotes the weight value for Y_{L-1}. In the generation of the sixth layer, N is 768, and j changes from 1 to 64. In the last-layer generation, N is 64, and j changes from 1 to 24. The softmax function is used for extracting the final output vector that indicates to which class the input frame cube belongs. The cost function for the training of the proposed CNN is defined using the cross entropy as follows:

$$C = -\sum_j^{24} y_j \times \log\left(\hat{Y}_7^j\right) \quad (3)$$

where j is an index for the output data; y and $\hat{Y}_7$ denote the ground-truth and final result of the proposed CNN, respectively. Once the training is completed, the proposed CNN produces a 24 × 1 output vector, $\hat{Y}_7$ for a given frame cube during the inference process. Subsequently, this output vector is used as the result of three consecutive input frames [$(t-1)$th, tth, $(t+1)$th input frames], as shown in Table I. Specifically, $\hat{Y}_7$ is divided into three 8 × 1 vectors ($\hat{Y}_7^R$), and $\hat{Y}_7^R$ is used as the output for each input frame. Thus, as mentioned in Section II-A1, the stride size of the input frame cube generation should be set to three. Each element in the 8 × 1 vector represents the probability that a specific event will occur among the eight counting events shown in Table II. Therefore, an element having the maximum value among the elements of the vector is derived, and an event indicated by the derived element is set as the final class of a given input frame cube. Using this extracted class, the proposed CNN can derive the instantaneous people count at the current time t (the current input frame cube).

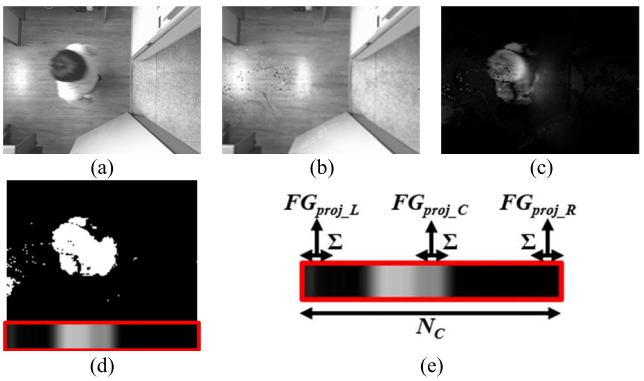

Fig. 6. FG detection and distribution analysis. (a) Input frame. (b) BG. (c) Δ_{BG_FG}. (d) FG with the projected FG on the x-axis (box with the red line, FG_{proj}). (e) FG distribution (FG_{proj_L}, FG_{proj_C}, FG_{proj_R}).

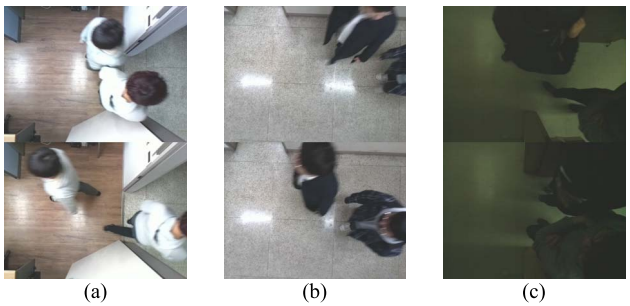

Fig. 7. Examples of test data set. (a) Example frames in DU1. (b) Example frames in DU2. (c) Example frames in Sogang University (SU).

TABLE IV
CASES OF COUNTING MODIFICATION

Conditions		
(Y_{max}^{1st}-Y_{max}^{2nd}) < TH_Y	$N^{IN} = 0$	($FG_L(t) > TH_H$ or $FG_L(t+1) > TH_H$) and $FG_C(t-1) > TH_H$
	$N^{IN} > 0$	$FG_L(t) < TH_L$ and $FG_L(t+1) < TH_L$
	$N^{OUT} = 0$	($FG_R(t) > TH_H$ or $FG_R(t+1) > TH_H$) and $FG_C(t-1) > TH_H$
	$N^{OUT} > 0$	$FG_L(t) < TH_L$ and $FG_L(t+1) < TH_L$

B. PCC Using FG Distribution With Event Probability

Based on the proposed CNN, it is possible to derive an instantaneous people count with better accuracy compared to conventional methods. However, if the training data are insufficient to reflect all the possible counting situations, the problem of overfitting occurs, which can decrease the counting accuracy of the proposed CNN. Thus, the counting results (resulting class) obtained by the proposed CNN were modified using the prior knowledge related to people counting and $\hat{Y}_7^R$ indicates the probability that a particular event had occurred. For utilizing prior knowledge, the BG of a given frame is first derived as follows:

$$\mathrm{BG}(x,y,t) = \begin{cases} \mathrm{BG}(x,y,t-1) & \text{if } \Delta(x,y,t) < 1 \\ L(x,y,t) & \text{otherwise} \end{cases}$$
$$\Delta(x,y,t) = |L(x,y,t) - L(x,y,t-1)| \quad (4)$$

where x and y are indexes for a spatial position, t is the index for a time of an input frame, and L denotes the luminance value of an input frame. In our method, L was calculated by averaging the R, G, and B color planes. The BG shown in Fig. 6(b) is the extracted BG. As shown in (4), the pixel in BG is updated when no change occurs in the pixel value on successive input frames. At the beginning of the input sequence, BG is set to the first input frame. After BG is calculated, the FG is extracted as follows:

$$\mathrm{FG}(x,y,t) = \begin{cases} 1, & \text{if } \Delta_{\mathrm{BG\_FG}}(x,y,t) > \mathrm{TH_{FG}} \\ 0, & \text{otherwise} \end{cases}$$
$$\Delta_{\mathrm{BG\_FG}}(x,y,t), = |\mathrm{BG}(x,y,t) - L(x,y,t)| \quad (5)$$

where TH_{FG} is the threshold value for the FG extraction and is determined by averaging the pixel values with values greater than 1 in Δ_{BG_FG}. Therefore, the TH_{FG} value can be adaptively determined depending on the input frame. Fig. 6(c) and (d) shows the Δ_{BG_FG} and FG, respectively. The extracted FG is projected on the x-axis [FG_{proj}, as shown in Fig. 6(d)] and is normalized as follows:

$$\mathrm{FG_{proj}}(x,t) = \sum_y \mathrm{FG}(x,y,t)$$
$$\mathrm{FG_{proj\_N}}(x,t) = \frac{\mathrm{FG_{proj}}(x,t)}{P_{\max} + \mathrm{offset}} \quad (6)$$

where P_{max} is the maximum value of FG_{proj}, and the offset in (6) was set to ten empirically. After the generation of the normalized FG_{proj} (FG_{proj_N}), the distribution of FG is analyzed as follows:

$$\mathrm{FG}_L(t) = \sum_{x=1}^{N_C/8} \mathrm{FG_{proj\_N}}(x,t)$$
$$\mathrm{FG}_C(t) = \sum_{x=N_C/2-N_C/16+1}^{N_C/2+N_C/16} \mathrm{FG_{proj\_N}}(x,t)$$
$$\mathrm{FG}_R(t) = \sum_{x=N_C-N_C/8+1}^{N_C} \mathrm{FG_{proj\_N}}(x,t) \quad (7)$$

TABLE V
NUMBER OF CLASS OCCURRENCES IN EACH DATA SET

Dataset (set number)		Class								
		1	2	3	4	5	6	7	8	Total
Training set	DU1(1~5)	1160	66	5	0	69	4	0	3	1307
	DU2(6~12)	2536	140	2	1	131	2	2	4	2818
	SU(13~21)	1573	121	0	0	116	0	0	7	1817
Total		5269	327	7	1	316	6	2	14	5942
Test set	DU1(1~10)	3245	138	3	0	137	4	0	5	3532
	DU2(11~18)	2366	136	1	2	132	1	1	2	2641
	SU(19~26)	1324	109	0	0	108	0	0	11	1552
Total		6935	383	4	2	376	6	1	18	7725

TABLE VI
IMPROVEMENTS IN COUNTING PERFORMANCE BY DA AND PCC

Error	Before DA	After DA	Improvements
E_{IN} ($\lvert N^{IN}$-$N^{IN}_{GT}\rvert$)	41	33	-8
E_{OUT} ($\lvert N^{OUT}$-$N^{OUT}_{GT}\rvert$)	42	33	-9
E (E_{IN}+E_{OUT})	83	66	-17 (79.5%)
Error	Before PCC	After PCC	Improvements
E_{IN}	33	24	-9
E_{OUT}	33	26	-7
E	66	50	-16 (75.8%)

Total N_{GT}^{IN}: 415, total N_{GT}^{OUT}: 409

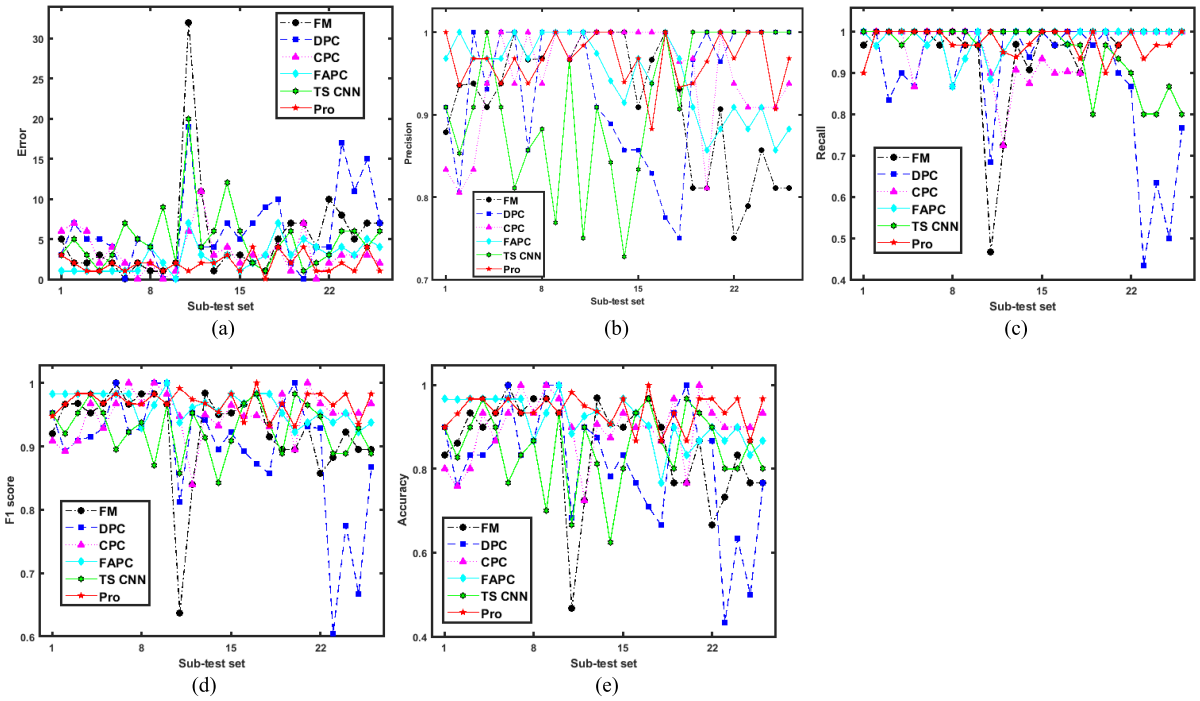

Fig. 8. Comparisons of the results of the benchmark of people counting and the proposed methods on all subtest image sets. (a) *E*. (b) Precision. (c) Recall. (d) F1 score. (e) Accuracy.

TABLE VII

ERRORS OF BENCHMARK AND PROPOSED METHODS ON TEST SETS (E_{IN}: $|N^{\text{IN}} - N_{\text{GT}}^{\text{IN}}|$, E_{OUT}: $|N^{\text{OUT}} - N_{\text{GT}}^{\text{OUT}}|$, E: $E_{\text{IN}} + E_{\text{OUT}}$)

Test sets	Ground-truth		FM			DPC			CPC			FAPC			TS CNN			Pro		
(subtest set)	$N^{IN}{}_{GT}$	$N^{OUT}{}_{GT}$	E_{IN}	E_{OUT}	E	E_{IN}	E_{OUT}	E	E_{IN}	E_{OUT}	E	E_{IN}	E_{OUT}	E	E_{IN}	E_{OUT}	E	E_{IN}	E_{OUT}	E
DU1 (1~10)	149	150	6	14	20	17	16	33	16	14	30	9	4	**13**	19	23	42	9	8	17
DU2 (11~18)	146	139	31	27	58	25	40	65	20	16	36	15	13	28	27	28	55	7	10	**17**
SU (19~26)	120	120	27	28	55	29	31	60	14	7	21	15	16	31	17	17	34	8	8	**16**
Total	415	409	64	69	133	71	87	158	50	37	87	39	33	72	63	68	131	24	26	**50**

where N_C is the number of pixels in $\text{FG}_{\text{proj\_}N}$. FG_L, FG_C, and FG_R shown in Fig. 6(e) represent the probabilities of existence of the FG in the left, right, and center positions of a given input frame, respectively. Subsequently, when the difference between the maximum value ($Y_{\text{max}}^{\text{1st}}$) and the second-largest value ($Y_{\text{max}}^{\text{2nd}}$) is $\hat{Y}_7^R$, the output class is adjusted based on the FG distribution (FG_L, FG_C, and FG_{R}). In other words, when the probability of the selected class (event) for a given input is not clearly higher than the next highest probability of the class, the resultant class is changed into the class having the $Y_{\text{max}}^{\text{2nd}}$ depending on the FG distribution, as described in Table IV. As shown, if the proposed CNN determines that no person leaves to the left ($N^{\text{IN}} = 0$), but the FG is present at the left of the current or next frame ($\text{FG}_L(t) > \text{TH}_H$ or $\text{FG}_L(t+1) > \text{TH}_H$) and at the center of the previous frame ($\text{FG}_C(t-1) > \text{TH}_H$), the counting result is changed to the event indicated by $Y_{\text{max}}^{\text{2nd}}$. If the proposed CNN determines that people leave to the left ($N^{\text{IN}} > 0$), but no FG exists on the left of the current and next frames ($\text{FG}_L(t) < \text{TH}_L$ and $\text{FG}_L(t+1) < \text{TH}_L$), the counting result is corrected to the event indicated by $Y_{\text{max}}^{\text{2nd}}$. To determine N^{OUT}, the same rules as described above are applied. TH_H, TH_L, and TH_Y were empirically set to 0.4, 0.1, and 15, respectively. The performance improvements by the PCC will be analyzed in Section III.

III. EXPERIMENTAL RESULTS

Simulations were conducted with three types of test sets (DU1, DU2, and SU) that were captured by the Raspberry Pi and Pi camera from Daegu and Sogang universities, as shown in Fig. 7. Specifically, DU1, DU2, and SU contain ten, eight, and eight subtest sets, respectively. Each test set is of the same capturing location and height. Each subset is a video captured by the individual launches of the Raspberry Pi, and thus, it has a different capturing time. Table V shows the number of subsets in training and testing (validation) sets and the number of class occurrences for each data set. As shown in this table, a total of 5492 frames were used for training, and a total of 7725 frames were used for testing. The total number of the in and out counts ($N_{\text{GT}}^{\text{IN}}$ and $N_{\text{GT}}^{\text{OUT}}$) in the test sets was 415 and 409, respectively. Each subtest set contains 15–30 in and out counts. The spatial and temporal resolutions of the test sets are 640 pixels × 480 pixels and 5 frames/s, respectively. The collected test sets include various types of entrance/exit situations of people of a target area (successive entrance/exit, entrance/exit of group consisting of two to three persons, and people crossing) and were produced in three illumination conditions (DU1, DU2, and SU) using the Raspberry Pi with Pi camera installed at various heights, as in [14].

For the training of parameters in the proposed CNN people counter, an Adam solver [22] was used. The initial step size

TABLE VIII

AVERAGE (AVG.), PRECISION (PRE.), RECALL (REC.), F1 SCORE (F1), AND ACCURACY (ACC.) WITH MINIMUM (MIN.) F1 AND ACC. OF BENCHMARK AND PROPOSED METHODS ON TEST SETS

Method	Test sets	Avg. Pre.	Avg. Rec.	Avg. F1	Avg. Acc.	Min F1	Min Acc.
FM	DU1	0.95	0.99	0.97	0.93	0.92	0.83
	DU2	0.98	0.86	0.90	0.85	0.64	0.47
	SU	0.82	1.00	0.90	0.77	0.86	0.67
	Avg.	0.92	0.95	0.92	0.85	-	-
DPC	DU1	0.95	0.95	0.94	0.89	0.89	0.76
	DU2	0.86	0.95	0.89	0.78	0.81	0.67
	SU	0.99	0.76	0.84	0.75	0.60	0.43
	Avg.	0.93	0.89	0.89	0.81	-	-
CPC	DU1	0.93	0.99	0.95	0.90	0.89	0.76
	DU2	1.00	0.88	0.93	0.88	0.84	0.73
	SU	0.92	1.00	0.96	0.91	0.90	0.77
	Avg.	0.95	0.96	0.95	0.90	-	-
FAPC	DU1	0.98	0.97	0.98	0.96	0.93	0.87
	DU2	0.96	0.98	0.97	0.90	0.94	0.77
	SU	0.89	1.00	0.94	0.87	0.92	0.83
	Avg.	0.94	**0.98**	0.96	0.91	-	-
TS_CNN	DU1	0.87	1	0.93	0.85	0.88	0.73
	DU2	0.86	1	0.92	0.82	0.84	0.63
	SU	1	0.83	0.91	0.83	0.89	0.8
	Avg.	0.91	0.94	0.92	0.83		
Pro	DU1	0.96	0.98	0.97	0.94	**0.95**	**0.90**
	DU2	0.96	0.97	0.97	0.93	0.93	**0.87**
	SU	0.97	0.97	0.97	0.93	**0.93**	**0.87**
	Avg.	**0.96**	0.97	**0.97**	**0.93**	-	-

TABLE IX

COMPARISON OF THE NUMBER OF WEIGHTS IN THE TS_CNN AND THE PROPOSED METHOD

Method	The number of weights	
	Without fully connected layer	Total
TS_CNN	(11×11×1×96+5×5×96×256 +3×3×256×384+3×3×384×384 +3×3×384×256) ×2=**7445184**	(11×11×1×96+5×5×96×256 +3×3×256×384+3×3×384×384 +3×3×384×256 +4608×4096+4096×4096+4096×1) ×2=**78756544**
Proposed	1×1×5×96+3×3×96×1 +(1×1×96×96+3×3×96×1) ×3 +3×3×96×1 = **32448**	1×1×5×96+3×3×96×1 +(1×1×96×96+3×3×96×1) ×3 +3×3×96×1 +768×64+64×24 = **83136**

for each training iteration was set to 2×10^{-3}, and it was decreased to 9/10 for every 3000 iterations. The training was terminated when the loss function defined in (3) had stopped decreasing.

For the benchmark methods, FM [6], directional people counter (DPC) [3], counting people crossing a line (CPC) [9], FAPC [14], and TS_CNN [17], which are the most popular methods for people counting, were used. All of the benchmark methods were implemented using MATLAB or Python or Tensorflow. For CPC, motion segmentation was performed using C language that is distributed by the author, and the remainder of the processing was implemented using MATLAB. For the training of TS_CNN, TSIs were generated using the same training data as the proposed method and used for its training. Adjustable parameters were optimized to produce the highest accuracy based on extensive experiments for a fair comparison.

For the evaluations of the benchmark and the proposed methods, the absolute difference between N^{IN} ($^{\mathrm{OUT}}$) and the ground-truth number of people entering or leaving the LOI ($N_{\mathrm{GT}}^{\mathrm{IN}}$ or $N_{\mathrm{GT}}^{\mathrm{OUT}}$) was used. In addition, the precision (Pre.), recall (Rec.), and F1 score (F1) were used for the following evaluations

$$\begin{aligned} \text{Pre.} &= \frac{\min\left(N^{\mathrm{IN}}, N_{\mathrm{GT}}^{\mathrm{IN}}\right) + \min\left(N^{\mathrm{OUT}}, N_{\mathrm{GT}}^{\mathrm{OUT}}\right)}{N^{\mathrm{IN}} + N^{\mathrm{OUT}}} \\ \text{Rec.} &= \frac{\min\left(N^{\mathrm{IN}}, N_{\mathrm{GT}}^{\mathrm{IN}}\right) + \min\left(N^{\mathrm{OUT}}, N_{\mathrm{GT}}^{\mathrm{OUT}}\right)}{N_{\mathrm{GT}}^{\mathrm{IN}} + N_{\mathrm{GT}}^{\mathrm{OUT}}} \\ F1 &= 2 \times ((\text{Rec.} \times \text{Pre.})/(\text{Rec.} + \text{Pre.})). \end{aligned} \tag{8}$$

The accuracy (Acc.) that was used in a previous study [6] was also used for the following evaluation:

$$\text{Acc.} = 1 - \frac{|N^{\mathrm{IN}} - N_{\mathrm{GT}}^{\mathrm{IN}}| + |N^{\mathrm{OUT}} - N_{\mathrm{GT}}^{\mathrm{OUT}}|}{N_{\mathrm{GT}}^{\mathrm{IN}} + N_{\mathrm{GT}}^{\mathrm{OUT}}}. \tag{9}$$

Before comparing the counting performances of the proposed method and the benchmark methods, the performance improvements by training DA and PCC were evaluated. Table VI shows the counting errors (E_{IN} and E_{OUT} corresponding to $|N^{\mathrm{IN}} - N_{\mathrm{GT}}^{\mathrm{IN}}|$ and $|N^{\mathrm{OUT}} - N_{\mathrm{GT}}^{\mathrm{OUT}}|$, respectively) before and after applying the DA and PCC. The total $N_{\mathrm{GT}}^{\mathrm{IN}}$ and $N_{\mathrm{GT}}^{\mathrm{OUT}}$ are shown at the bottom of Table VI, representing the total number of people entering or leaving the target place of all subtest sets. As shown in this table, the counting error was reduced to 79.5% through DA and could be reduced further to 75.8% through PCC. In fact, it is impossible to create training data pairs that include all possible entrance and exit situations. Therefore, it is highly probable that the counting performance of the proposed CNN will be deteriorated owing to the overfitting problem. In fact, the proposed CNN without DA and PCC causes frequent counting errors for the test sets owing to this overfitting problem. As shown in Table VI, DA and PCC are fundamental in mitigating the degradation of counting performance due to the overfitting problem.

Next, the counting performances of the proposed method using both DA and PCC with the benchmark methods are compared. To extract the final cumulative count, the benchmark methods, except for TS_CNN, utilized the regression process using the ground-truth cumulative count for each subtest set, while the TS_CNN and the proposed method extracted the final cumulative count by summing the instantaneous count without any regression process. Table VII shows the E_{IN}, E_{OUT}, and E. The detailed results and errors for each subtest set are shown in Fig. 8(a). Among the benchmark methods, DPC indicated the highest E_{IN} and E_{OUT} values, whereas FAPC indicated the lowest values. Since the regression process was not applied to the TS_CNN, the E value of TS_CNN was larger than that of the FAPC or CPC. Compared with FAPC, the proposed method indicated the lower E_{IN} and E_{OUT} values for all test sets except DU1. The proposed method reduced the total E to 67.9% that of FAPC. Compared with TS_CNN that utilizes TSIs for the people counting, the proposed method provided lower errors for all test sets. In an indoor environment with a very low frame rate of the input, the counting accuracy of a TSI-based method may be degraded because it is difficult for the FG information to be fully reflected in the TSI and it is very difficult to accurately derive the motion

information (optical flow) of the FG. For all subtest sets, the proposed method provided a robust counting performance with no noticeable error, whereas the benchmark methods produced extreme errors for some subtest sets, as shown in Fig. 8(a). Fig. 8(b)–(d) shows the Pre., Rec., F1, and Acc. for all subtest sets. In Table VIII, the average and minimum Pre., Rec., F1, and Acc., values for each test set are shown. As shown in these figures and tables, the proposed method indicated the best Pre., Rec., and F1 values in almost all subtest sets. Similarly, in the comparison of Acc., the proposed method provided the best results for all test sets. Specifically, the proposed method provided the best average F1 and Acc., values for all test frames. Although the proposed method provided better counting accuracy compared with benchmark methods, it was confirmed that the double-counting errors that recognize a single entrance/exit event as two consecutive entrance/exit events occur in some frames (18 frames out of a total of 7725 frames in test data set, 0.23%) due to an inaccurate event classification.

Finally, to compare the computational complexities of the TS_CNN and the proposed method, the total numbers of weights used for each network were compared, as shown in Table IX. Although it is difficult to accurately compare the computational complexities of the two methods because the sizes of the input data for each layer are different, it can be seen that the proposed method uses a much lower number of weights compared to those of the TS_CNN.

IV. Conclusion

In this article, a novel CNN-based approach for estimating people's count was proposed. In the proposed method, the problem of people counting was converted to that of event classification. Subsequently, the proposed CNN was trained using input frame cubes with their labeled classes. During the training, the augmentation of training data pairs was performed to alleviate the overfitting problem. After the training, the trained CNN classified a given input frame cube to a specific event that represents the number of people entering and leaving a target area. For improving the accuracy of people count, the FG distribution and probabilities of classes for a given input frame cube calculated by the proposed CNN were used.

The benefits of the proposed method were verified through various experiments using three test sets produced in various environments. In the experiments, the proposed method provided the best accuracy of people counting compared with the benchmark methods.

In our future work, I plan to evolve the proposed method to classify more complex entry and exit situations by creating additional event classes. In addition, I will expand training and testing data sets to consider more diverse people counting environments.

References

[1] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: The rising trend of vision based measurement," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 3, pp. 41–47, Jun. 2014.

[2] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian detection of independent motion in crowds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 594–601.

[3] J. García, A. Gardel, I. Bravo, J. L. Lázaro, M. Martínez, and D. Rodríguez, "Directional people counter based on head tracking," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3991–4000, Sep. 2013.

[4] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 705–711.

[5] A. G. Vicente, I. B. Munoz, P. J. Molina, and J. L. L. Galilea, "Embedded vision modules for tracking and counting people," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 9, pp. 3004–3011, Sep. 2009.

[6] Y. Cong, H. Gong, S.-C. Zhu, and Y. Tang, "Flow mosaicking: Real-time pedestrian counting without scene-specific learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1093–1100.

[7] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[8] Z. Ma and A. B. Chan, "Crossing the line: Crowd counting by integer programming with local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2539–2546.

[9] Z. Ma and A. B. Chan, "Counting people crossing a line using integer programming and local features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 10, pp. 1955–1969, Oct. 2016.

[10] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2871–2878.

[11] H. Xu, P. Lv, and L. Meng, "A people counting system based on head-shoulder detection and tracking in surveillance video," in *Proc. ICCDA*, vol. 1, Jun. 2010, pp. V1-394–V1-398.

[12] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1113–1121.

[13] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *Proc. BMVC*, 2012, p. 3.

[14] S. I. Cho and S.-J. Kang, "Real-time people counting system for customer movement analysis," *IEEE Access*, vol. 6, pp. 55264–55272, 2018.

[15] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 833–841.

[16] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4031–4039.

[17] L. Cao, X. Zhang, W. Ren, and K. Huang, "Large scale crowd analysis based on convolutional neural network," *Pattern Recognit.*, vol. 48, no. 10, pp. 3016–3024, Oct. 2015.

[18] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 712–726.

[19] L. Sifre, "Rigid-motion scattering for image classification," Ph.D. dissertation, Dept. Math. Appl., École Polytechnique Univ., Palaiseau, France, 2014.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

Sung In Cho (S'10–M'17) received the B.S. degree in electronic engineering from Sogang University, Seoul, South Korea, in 2010, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2015.

From 2015 to 2017, he was a Senior Researcher with LG Display, Gyeonggi-do, South Korea. From 2017 to 2019, he was an Assistant Professor of electronic engineering with Daegu University, Gyeongsan-si, South Korea. He is currently an Assistant Professor of multimedia engineering with Dongguk University, Seoul. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, and circuit design for liquid crystal display (LCD) and organic light-emitting diode (OLED) systems.