Received September 12, 2019, accepted October 27, 2019, date of current version December 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953558

Non-Temporal Lightweight Fire Detection Network for Intelligent Surveillance Systems

HUNJUN YANG [1], (Student Member, IEEE), HYEOK JANG [2], TAEYONG KIM [3], AND BOWON LEE [1], (Senior Member, IEEE)

[1] Department of Electronic Engineering, Inha University, Incheon 22212, South Korea
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea
[3] Hyundai Robotics, Yongin 16891, South Korea

Corresponding author: Bowon Lee (bowon.lee@inha.ac.kr)

This work was supported in part by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Korea Government (MOLIT) under Grant 19TLRP-B148386-02, Commercial Vehicle-Based Road and Traffic Information System, and in part by the Industrial Technology Innovation Program funded by the Ministry of Trade, Industry and Energy (MI, Korea) under Grant 10073154.

ABSTRACT Convolutional neural networks (CNNs) have been recently applied to tackle a variety of computer vision problems. However, because of its high computational cost, careful considerations are required to design cost-effective CNNs. In this paper, we propose a CNN inspired by MobileNet for fire detection in surveillance systems. In the proposed network, color features emphasized by the channel multiplier are extracted through depthwise separable convolution, and squeeze and excitation modules further increase the representation of the channel-wise convolution. Custom Swish is used as an activation function to limit exceedingly high weights from the effects of the channel multiplier. Our proposed network achieves 95.44% accuracy for fire detection, which is higher than those achieved other existing networks. Furthermore, the number of parameters used is 38.50% fewer than that of MobileNetV2, the smallest among other networks. We believe that using the proposed CNN, CNN-based surveillance systems could be implemented in lightweight devices without using expensive dedicated processors.

INDEX TERMS Fire detection, deep learning, convolutional neural networks, image classification.

I. INTRODUCTION

Fires can occur anywhere, at any time, and if they are not detected early, they can cause severe damages to property and people. Surveillance systems consisting of multiple CCTVs can be very useful in detecting fires because they are designed to monitor the surroundings 24 hours a day. Furthermore, they can be very useful to monitor fires in a wide range of areas, including inaccessible areas. Consequently, there has been a huge demand for intelligent video-based fire monitoring systems that can alert people to respond quickly by processing and analyzing video streams in real-time. A video-based fire detection system can inform an operator by analyzing videos from CCTVs without using heat, smoke, or flame sensors. Owing to the significant development of video analysis, video signals from CCTVs can be automatically analyzed and can provide alarms to surveillance personnel to enable quick response.

Traditional vision-based fire detection methods use handcrafted features, such as color, motion, and texture. Prior studies [1]–[4] detected fire by making full use of color features because fire is generally brighter and has higher contrast than other objects. Ko *et al.* [1] detected specific fire regions from their color and, then employed a model using wavelet coefficients to detect fire with a support vector machine (SVM) classifier. Celik *et al.* [5] extracted foreground information using color-based background modeling, and classified fires using a generic statistical model. Chen *et al.* [3] extracted RGB-based chromatic and disorder measurements for fire detection. Li *et al.* [4] proposed a flame detection framework based on the color, dynamics, and flickering properties of flames. They used a Dirichlet-process Gaussian mixture model-based approach for autonomous flame detection.

Because color information is often influenced by environmental conditions, motion information is often used as a feature. Kuo *et al.* [6] detected flame boundaries by using motion detection. In their work, a motion mask was used for motion detection, and flames were detected through the analysis

The associate editor coordinating the review of this manuscript and approving it for publication was Vladimir M. Mladenovic.

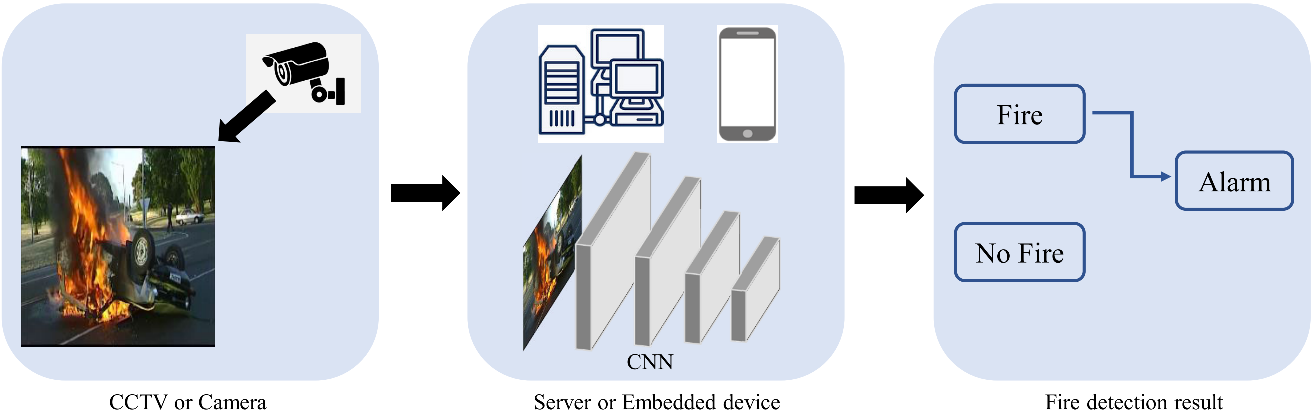


FIGURE 1. **Vision-based fire detection system using convolutional neural networks.**

of the flame behavior. Ko *et al.* [7] proposed an adaptive background subtraction model with hierarchical Bayesian networks. By analyzing and modeling fire and fire-like moving objects, they improved the detection performance. Habiboğlu *et al.* [8] used texture information as a feature. They divided video data into spatio-temporal blocks and extracted covariance-based features. Then, they trained the extracted features with an SVM classifier to detect fire.

In computer vision, numerous convolutional neural networks (CNNs) have been developed since AlexNet [9] and VGG [10] proved that the higher non-linearity with deeper network could improve the performance significantly. Since then, researchers have created deeper CNN structures to improve the performance further. Unfortunately, with deeper networks, vanishing/exploding gradient and degradation problems can arise [11]–[13]. To solve these problems, various techniques, such as batch normalization (BN) [14], Dropout [15], and various activation functions have been applied to the network. In particular, ResNet [12] and GoogleNet [11] were able to build deep networks by using residual and inception modules as basic blocks, respectively. They demonstrated better performance in the ILSVRC [16] and have been continuously explored through the deformation block [17]–[19]. ResNet is widely used as a base network not only for classification but also for detection and segmentation with localization [20]–[23].

Since the processors of embedded hardware have become more powerful, there have been continuous demands for lightweight and high performance networks that can operate on embedded systems. Xception [24] and MobileNet [25], [26] use lighter depthwise separable convolutions compared with standard convolutions, reducing parameters so that networks can be used on mobile devices. ShuffleNet [27], [28] reduced the computational cost using pointwise group convolution and channel shuffle.

Traditional fire detection methods require hand-crafted features. In contrary, CNN has been proven to work significantly better than traditional methods without requiring a feature extraction stage because the features are learned through the network. In recent years, various CNN-based fire detection methods have been explored [29]–[33]. Zhong *et al.* [29] used the RGB model to detect candidate flame regions and classify them with CNNs to detect flames. Frizzi *et al.* [30] detected fire using a 9-layer CNN. Dunnings and Breckon [31] proved that modified AlexNet and InceptionV1 achieved better performance than existing CNNs. Muhammad *et al.* [32] proposed a CNN architecture inspired by GoogleNet and demonstrated better performance than hand-crafted feature-based algorithms. Muhammad *et al.* [33] proposed a MobileNet-based architecture for fire detection with a cost-efficient CNN.

Figure 1 shows a diagram of a vision-based fire detection system using CNNs. The video obtained by the CCTV or camera is classified using CNN from a server or embedded device. If the input is classified as fire, an alarm can be triggered. CNN architectures generally require high computational cost and memory usage. In order to use CNNs more effectively for image analysis, we propose a network for fire detection using a channel multiplier and squeeze and excitation (SE) depthwise modules, along with a modified version of the Swish activation function. Our experimental results demonstrated that the proposed method achieves higher accuracy than the recently proposed MobileNetV2 [26], despite having fewer network parameters and a lower computational cost.

II. PROPOSED ALGORITHM

In this section, we describe our non-temporal lightweight fire detection method using CNN. Because a fire object has different characteristics from other objects, especially color, we propose the use of a channel multiplier to emphasize the color features in our network. We also use SE-Depthwise modules to enhance the representation of depthwise separable convolution for effectively representing the global features of fire, because fire, in general, is a deformable object. Furthermore, although conventional CNNs may show high classification performance, inferences tend to take a considerable amount of time because of their large number of parameters.

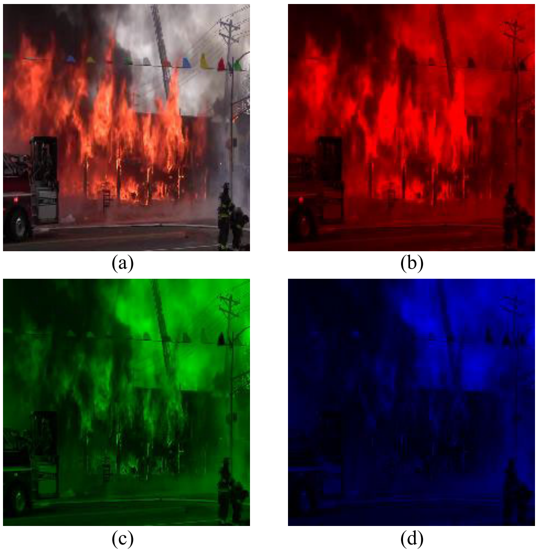

(a) (b) (c) (d)

FIGURE 2. A sample fire image, (a) fire image, (b) red channel, (c) green channel, (d) blue channel.

We address this issue by reducing the complexity of the network so that CNN can be used for embedded systems with lower computational power.

A. FIRE OBJECT CHARACTERISTICS

A fire object has significantly different characteristics compared with other objects, such as a car, person, or table, which have a relatively consistent shape and size. In particular, a fire object is deformable, and some of its features, such as color and texture have consistent characteristics compared to its shape and size. Therefore, these features can be useful for fire detection tasks.

Figure 2 shows a sample fire image along with its red, green, and blue channels. As shown in figure 2 (a), although the fire is deformable, it shows relatively constant color and texture patterns. In Figure 2 (b), the shape of the fire is better represented compared with those in Figure 2 (c) and (d). Therefore, we consider that the fire features can be extracted more effectively when the red channel is enhanced.

B. CHANNEL MULTIPLIER

We propose using a *channel multiplier* on depthwise separable convolution to emphasize the color characteristics of the fire. We selected depthwise separable convolution because it has a significantly lower computational cost. Figure 3 shows the process of depthwise separable convolution. Unlike the general convolution process, it is divided into two processes: depthwise convolution and pointwise convolution. Depthwise convolution divides the input by channels, performs convolution with kernels in each channel, and then stacks the output to create a feature map. Pointwise convolution uses 1 × 1 kernels for the output generated in the depthwise convolution procedure. The 1 × 1 kernel can be used to increase or decrease the size of the output feature map. Depthwise separable convolution can significantly lower the model complexity compared with standard convolution.

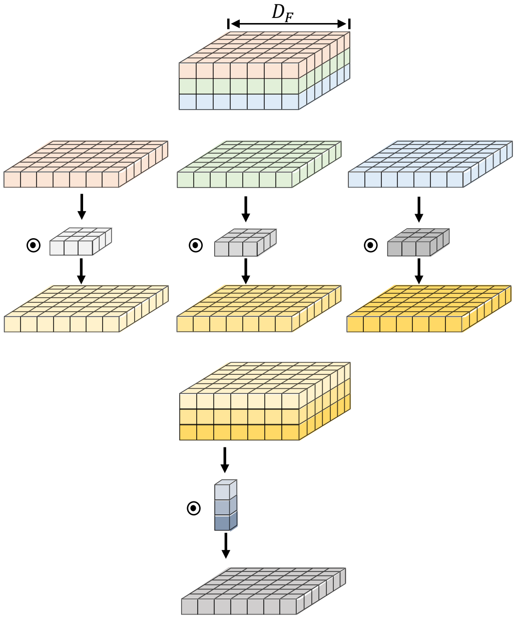


FIGURE 3. Depthwise separable convolution.

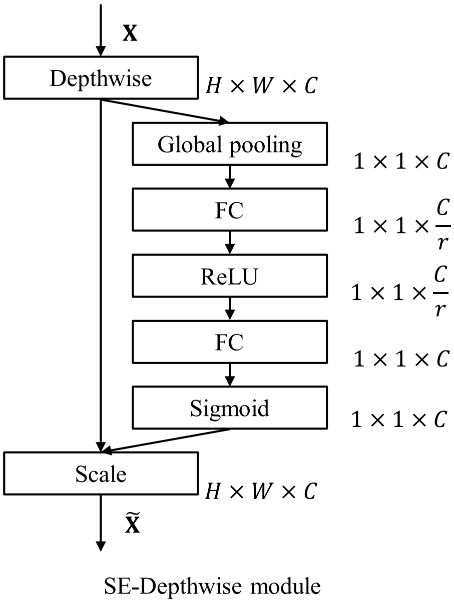


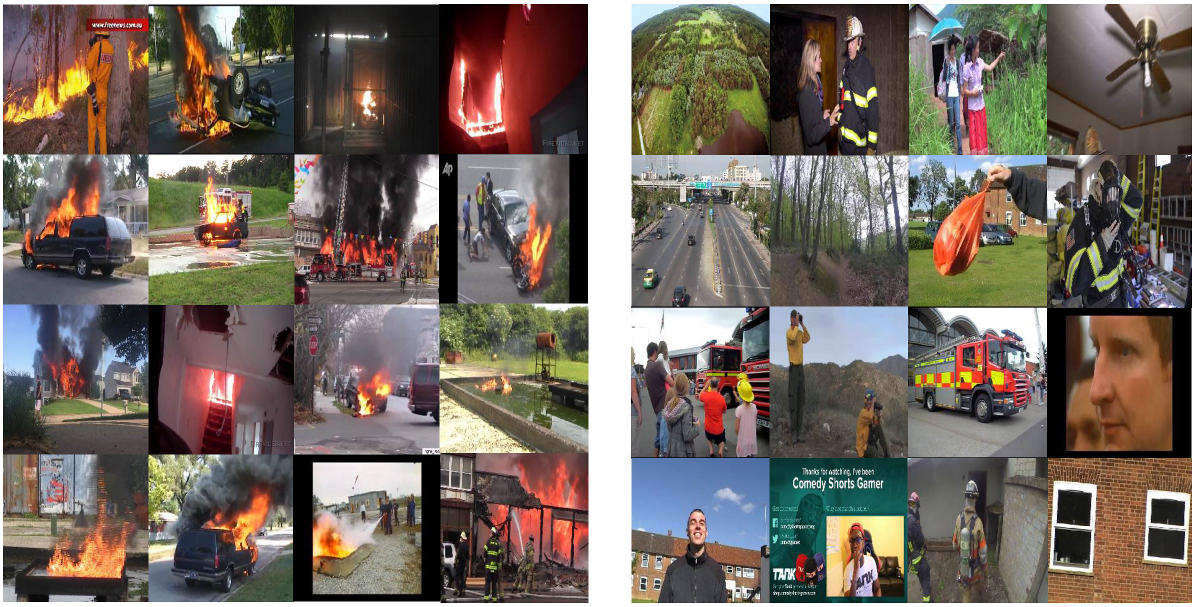
FIGURE 4. A Diagram of the SE-depthwise module.

In the process of depthwise convolution, a feature map $\mathbf{F}$ with dimensions $D_F \times D_F \times N$ is produced when the input dimension is $D_F \times D_F \times M$ where M and N are the numbers of input and output channels, respectively, and D_F is the width and height of the input.

Depthwise convolution can be represented as follows:

$$\hat{\mathbf{G}}_{k,l,m} = \sum_{i,j} \mathbf{a} \cdot \hat{\mathbf{K}}_{i,j,m} \cdot \mathbf{F}_{k+i-1,l+j-1,m}, \tag{1}$$

where $\hat{\mathbf{K}}$ is the depthwise convolutional kernel, $\hat{\mathbf{G}}$ is the output feature map, and $\mathbf{a}$ is the channel multiplier. The channel multiplier $\mathbf{a}$ can be represented as $[a_r, a_g, a_b]$ where a_r, a_g, a_b are non-negative real numbers.



(a) (b)

FIGURE 5. Sample dataset images, (a) fire image, (b) non-fire image.

Because the input image has three channels (RGB), the channel multiplier elements $[a_r, a_g, a_b]$ are multiplied with the corresponding channels. With the experiment results, we will show that the channel multiplier is useful in extracting specific color patterns of objects.

C. SE-DEPTHWISE MODULE

Convolution kernels are designed to learn local receptive fields of feature maps rather than the global perspective. By integrating the local fields, nonlinear relationships can be inferred and global features can be provided through pooling. SENet [34] implements the squeeze and excitation (SE) technique to recalibrate the feature map of the network and performs better compared with existing networks, such as VGGNet, Inception, and ResNet.

Squeeze can be obtained in the process of global average pooling. Let $\mathbf{X}$ and $\tilde{\mathbf{X}}$ be the input and output of an SE block, respectively, and u_c be the entry of a convolution result of $\mathbf{X}$. Then, the squeeze output z_c can be calculated as in Equation (2), where the spatial dimension is $H \times W$.

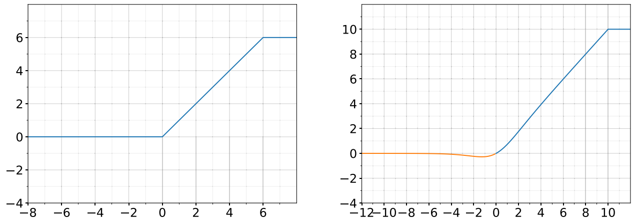
$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j), \tag{2}$$

where $\mathbf{z} \in R^C$.

In the process of excitation, the excitation output is

$$\mathbf{s} = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})), \tag{3}$$

where δ is the ReLU [35] function, and σ is the sigmoid function. $\mathbf{W}_1 \in R^{\frac{C}{r} \times C}$ and $\mathbf{W}_2 \in R^{C \times \frac{C}{r}}$ are the weights of fully connected layers where r is the reduction ratio.



(a) (b)

FIGURE 6. Activation functions, (a) ReLU6, (b) Swish10.

Figure 4 shows a diagram of the SE-Depthwise module. In SENet [34], the SE block can be easily applied to various networks, such as SE-Inception and SE-ResNet. In our work, we create an SE-Depthwise module by applying an SE block after depthwise separable convolution with a reduction ratio r of 16. The SE-Depthwise module is effective for obtaining the characteristics of fire objects, which have variable sizes and shapes, because it can learn global properties from images. The SE-Depthwise module is used to build the main architecture except for the network input and output layers.

D. PROPOSED NETWORK

In general, CNN tends to perform better with deeper networks. However, when there are only a few classes, with simple characteristics, overfitting can occur. If the network is too deep, although overfitting does not occur, the performance of the network may not improve significantly despite the increased number of parameters. In our proposed network, the layers are designed to mitigate the overfitting problem and maximize the performance by adjusting the depth of the network and the number of feature maps for fire detection.

TABLE 1. MobileNet-fire architecture.

block	conv	stride	Filter Shape	Input size
DConv	dw conv	(2,2)	$3^2 \times 3$	$224^2 \times 3$
	pw conv	(1,1)	$1^2 \times 3 \times 32$	$112^2 \times 3$
SE-DConv	dw conv	(1,1)	$3^2 \times 32$	$112^2 \times 32$
	pw conv	(1,1)	$1^2 \times 32 \times 64$	$112^2 \times 32$
SE-DConv	dw conv	(2,2)	$3^2 \times 64$	$112^2 \times 64$
	pw conv	(1,1)	$1^2 \times 64 \times 128$	$56^2 \times 64$
SE-DConv	dw conv	(1,1)	$3^2 \times 128$	$56^2 \times 128$
	pw conv	(1,1)	$1^2 \times 128 \times 128$	$56^2 \times 128$
SE-DConv	dw conv	(2,2)	$3^2 \times 128$	$56^2 \times 128$
	pw conv	(1,1)	$1^2 \times 128 \times 256$	$28^2 \times 128$
SE-DConv	dw conv	(1,1)	$3^2 \times 256$	$28^2 \times 256$
	pw conv	(1,1)	$1^2 \times 256 \times 256$	$28^2 \times 256$
SE-DConv	dw conv	(2,2)	$3^2 \times 256$	$28^2 \times 256$
	pw conv	(1,1)	$1^2 \times 256 \times 256$	$14^2 \times 256$
SE-DConv	dw conv	(1,1)	$3^2 \times 256$	$14^2 \times 256$
	pw conv	(1,1)	$1^2 \times 256 \times 256$	$14^2 \times 256$
SE-DConv	dw conv	(1,1)	$3^2 \times 256$	$14^2 \times 256$
	pw conv	(1,1)	$1^2 \times 256 \times 256$	$14^2 \times 256$
SE-DConv	dw conv	(2,2)	$3^2 \times 256$	$14^2 \times 256$
	pw conv	(1,1)	$1^2 \times 256 \times 512$	$7^2 \times 256$
SE-DConv	dw conv	(1,1)	$3^2 \times 512$	$7^2 \times 512$
	pw conv	(1,1)	$1^2 \times 512 \times 512$	$7^2 \times 512$
-	Avg Pool	-	Pool 7×7	$7^2 \times 512$
-	FC	-	512×1024	$1^2 \times 1024$
-	sigmoid	-	classifier	$1^2 \times 2$

dw conv = depthwise convolution, pw conv = pointwise convolution

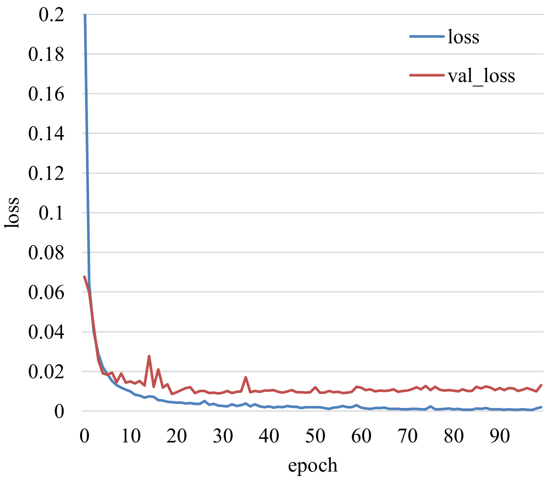


FIGURE 7. Training and validation losses.

Our proposed network consists of 11 blocks, as shown in Table 1. The input size of the network is 224 × 224 × 3. The first block of the network is the depthwise separable convolution (DConv) with the channel multiplier, as shown in Equation (1). In the DConv block, depthwise convolution is performed first with BN and activation. Then, pointwise convolution with BN and activation is performed. In this process, a 3 × 3 convolution kernel with stride (2, 2), and a channel multiplier $\mathbf{a} = [2.0, 1.0, 1.0]$ are used.

After the first block, SE-DConv blocks are stacked 10 times, as shown in Figure 4. Similar to DConv, SE-DConv consists of an SE block after depthwise and pointwise convolution with 3 × 3 kernels. In the SE-DConv blocks, downsampling with stride (2, 2) is performed four times. We selected the number of feature maps and the output channels of the final SE-DConv blocks as [32, 64, 128, 256, 512] and 7 × 7 × 512, respectively, because they achieved the best performance in our experiments. Finally, global average pooling is performed to obtain a spatial resolution of 1 × 1 and then, the number of output channels is increased to 1024 using a fully connected layer. As the classifier, the sigmoid function is employed.

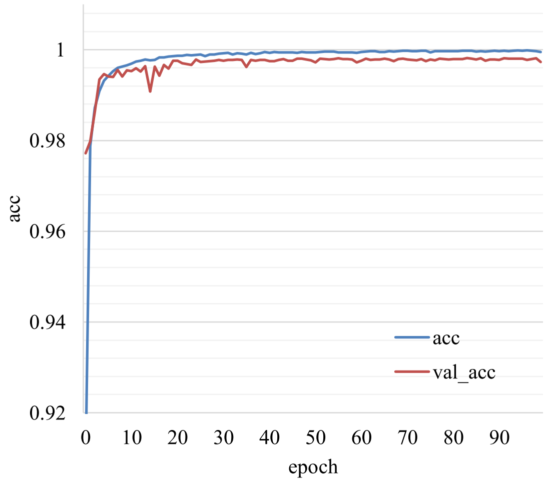


FIGURE 8. Training and validation accuracies.

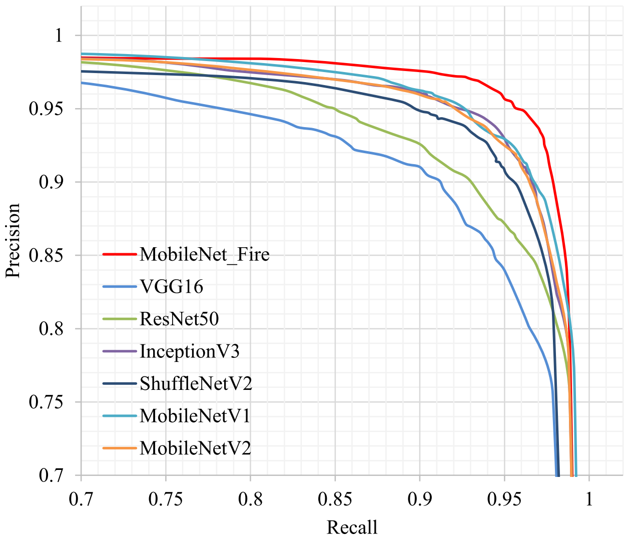


FIGURE 9. ROC curve comparison.

Various activation functions are tested to improve the performance of the network. In particular, ReLU was in prior studies owing to its better performance and reduced training speed. Various custom ReLUs [36], [37] and Swish [38] activation functions have been explored for further improvements. Figure 6 (a) shows the ReLU6 activation function used in MobileNet [25], [26]; this function limits the weight by clipping the positive range of the output at $a = 6$ in Equation (4).

$$f(x) = min(max(0, x), a). \tag{4}$$

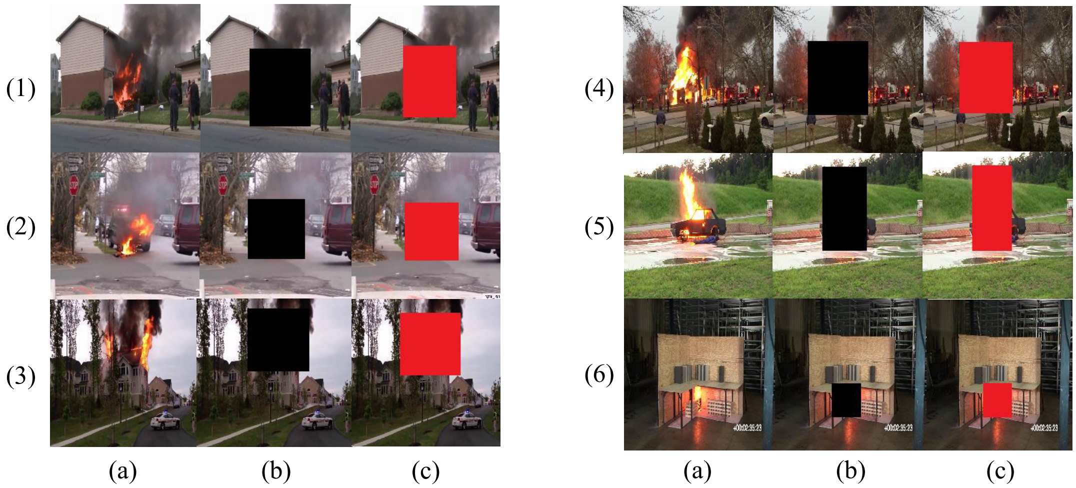


FIGURE 10. Fire image examples and visualization, (a) fire image, (b) black occlusion, (c) red occlusions.

Furthermore, the Swish activation function has been proven to be superior to ReLU [38] and has frequently been used as a replacement for ReLUs. Similar to the ReLU6 case, we further modify the Swish activation function by clipping its output, as shown in Equation (5).

$$f(x) = \begin{cases} min(\dfrac{x}{1+e^{-x}}, a), & x \geq 0 \\ \dfrac{x}{1+e^{-x}}, & x < 0. \end{cases} \quad (5)$$

Figure 6 (b) shows the Swish10 activation function with $a = 10$. In our proposed network, Swish10 is selected as the activation function to preserve lager values due to the effect of the channel multiplier.

III. EXPERIMENTS

This section presents the experimental results and a comparison of the proposed network with existing CNNs for fire detection. A multi-GPU system equipped with an E5-1650 v4 CPU with six cores and four NVIDIA Titan Xp GPUs are used for the experiments.

A. DATASET

For the experiments, datasets with various types of fire images were obtained from Chenebert *et al.* [39], Hüttner *et al.* [40], and Steffens *et al.* [41], [42]. Each video frame was resized to 224 × 224 before use. The total number of images was approximately 340 k, 80% and 20% of which were used for training and validation, respectively. Approximately 36 k images were used for testing, with a the ratio of fire to non-fire images of approximately 50:50. Figure 5 (a) and (b) displays examples of fire images and non-fire images from our customized dataset, respectively. The fire images are mainly composed of images that clearly show flames. Non-fire images include deformable objects, such as trees, grass, and the sky, as well as rigid objects, such as people and cars.

TABLE 2. Comparison of fire detection results.

Model	Accuracy	Million Mult-Adds	Million Parameters	Parameter Reduction
VGG16	90.46%	15,610	134.27	98.95%
ResNet50	91.31%	3,889	23.59	94.11%
InceptionV3	94.11%	5,745	21.81	93.63%
ShuffleNetV2	93.23%	491	4.02	65.42%
MobileNetV1	93.68%	571	3.23	56.97%
MobileNetV2	93.82%	302	2.26	38.50%
MobileNet-Fire	95.44%	263	1.39	-

B. TRAINING

Since the development of the stochastic gradient descent (SGD) optimizer, various other optimizers have been developed, such as Momentum [43] and NAG [44] considering the directionality, and Adagrad [45], RMS Prop, and AdaDelta [46] considering the step size. Among them, the Adam optimizer [47], which considers both directionality and step size, is the most frequently used.

To compare optimizers, Wilson *et al.* [48] examined their performance for a binary classification task. They verified that adaptive optimizers, such as the Adam optimizer, might achieve worse results than SGD. Similarly, we verified that the adaptive optimizers do not necessarily guarantee better performance. Therefore, our network was trained with SGD, with the momentum and learning rate set to 0.9 and 0.001, respectively. The batch size was 64, and the loss function used binary cross-entropy.

Figure 7 shows the training and validation losses and Figure 8 displays the training and validation accuracies, both up to 100 epochs. Because the loss and accuracy saturated after 50 epochs, increasing the epoch any further does not improve the performance significantly.

C. RESULTS

Table 2 shows the fire detection accuracy and number of parameters for existing CNNs and our proposed network,

TABLE 3. Fire detection result with Figure 10.

image	occlusion	MobileNet_Fire		VGG16		ResNet50		InceptionV3		ShuffleNetV2		MobileNetV1		MobileNetV2	
		nofire	fire	nofire	fire	nofire	fire	nofire	fire	nofire	fire	nofire	fire	nofire	fire
(1)	-	0	100	0	100	0	100	0	100	0	100	0	100	0	100
	black	99.9	0.1	89.75	10.25	99.95	0.05	100	0	99.95	0.05	98.85	1.15	100	0
	red	99.97	0.02	99.87	0.13	100	0	96.25	3.75	100	0	100	0	100	0
(2)	-	0.01	99.99	0	100	0	100	0	100	0	100	0	100	0	100
	black	100	0	99.99	0.01	100	0	100	0	100	0	100	0	100	0
	red	100	0	100	0	100	0	100	0	100	0	100	0	100	0
(3)	-	0	100	0	100	0	100	0	100	0	100	0	100	0	100
	black	99.91	0.09	99.99	0.01	0.38	99.62	99.72	0.28	0.21	99.79	98.12	1.88	23.22	76.78
	red	99.78	0.23	0.01	99.99	49.46	50.54	87.64	12.36	99.33	0.67	99.54	0.46	96.46	3.54
(4)	-	0.04	99.96	0.18	99.82	38.73	61.27	0	100	0	100	0	100	0.01	99.99
	black	99.47	0.53	4.89	95.11	100	0	44.12	55.88	100	0	98.25	1.75	95.94	4.06
	red	78.16	23.86	88.89	11.11	100	0	86.95	13.05	100	0	99.87	0.13	99.98	0.02
(5)	-	0	100	0	100	0	100	0	100	0	100	0	100	0	100
	black	99.57	0.44	0	100	95.26	4.74	0	100	99.23	0.77	69.23	30.77	55.48	44.52
	red	90.8	8.76	19.24	80.76	0.03	99.97	0	100	100	0	95.35	4.65	99.42	0.58
(6)	-	2.54	97.38	0	100	10.06	89.94	1.03	98.97	0	100	73.7	26.3	0.32	99.68
	black	99.95	0.05	0.01	99.99	100	0	100	0	99.91	0.09	100	0	100	0
	red	100	0	0.01	99.99	100	0	99.99	0.01	98.52	1.48	100	0	100	0
error count		0		6		3		3		1		1		1	

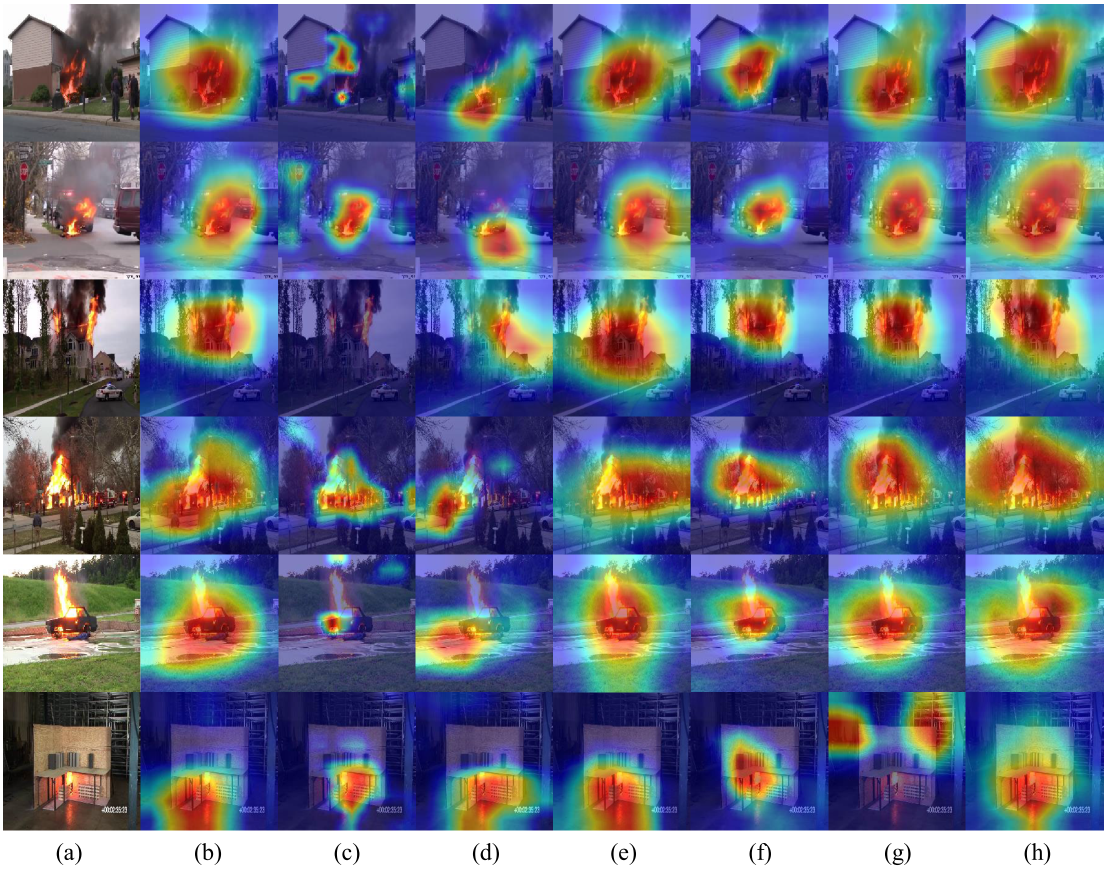


FIGURE 11. Grad-CAM localizations of fire detection models, (a) fire image, (b) MobileNet-Fire, (c) VGG16, (d) ResNet50, (e) InceptionV3, (f) ShuffleNetV2, (g) MobileNetV1, (h) MobileNetV2.

which was named "MobileNet-Fire". The parameter reduction in Table 2 is the ratio of reduced parameters in MobileNet-Fire compared with each network. Among the networks, VGG16 has the highest number or parameters (134.27 M) because two fully connected layers are used at the end of the network. MobileNet-Fire uses only 1.39 M parameters, which represents a reduction of 98.95 %, and has an accuracy increase of 4.98%. Compared with ResNet50 and InceptionV3, our network demonstrated a 4.13 % and 1.33 % higher accuracy, respectively, and 94.11 % and 93.63

TABLE 4. Detection results for different activation functions.

Model	Activation Function	Accuracy
MobileNet-Fire	ReLU6	91.99%
MobileNet-Fire	ReLU10	94.91%
MobileNet-Fire	Swish	93.54%
MobileNet-Fire	Swish6	93.42%
MobileNet-Fire	Swish10	95.44%

% fewer parameters, respectively. Compared with ShufflenetV2, the number of parameters of MobileNet-Fire is reduced by 65.42%, and the accuracy is increased by 2.21%. Compared with MobileNetV1 and MobileNetV2, which are considered lightweight networks, our network has 56.97% and 38.50% fewer parameters, respectively, and accuracy improvements of 1.76% and 1.62%, respectively.

Figure 9 shows the precision-recall ROC curves. The precision is the percentage of sampled images that the model classified as positive, that are actually positive, and the recall is the percentage of results correctly classified by the algorithm. Thus, precision and recall generally have a trade-off relationship. The performance of the model in distinguishing between classes can be analyzed from the ROC curve; a larger area under the ROC curve indicates better performance of the model. According to Figure 9 and Table 2, MobileNet-Fire achieved the best overall performance.

D. CHOICE OF ACTIVATION FUNCTION

The accuracies for different activation functions are displayed in Table 4. ReLU10 outperformed ReLU6, with an accuracy of 94.91 %. We suspect that larger values can be more effectively passed in ReLU10 owing to the effect of the channel multiplier. When the Swish and Swish6 activation functions were used, the model achieved accuracies of 93.54 % and 93.42 %, respectively. The highest accuracy, 95.44 %, was achieved by MobileNet-Fire when using Swish10, which was chosen as the activation function for our model.

IV. ROBUSTNESS ANALYSIS

Verifying the robustness of the system in various environments is crucial in computer vision. To demonstrate the robustness of our network, we tested the fire detection algorithm with selected images. This task can be confirmed by detecting the flame in the fire image. Figure 10 shows the 6 fire images selected from the dataset. Each image was processed with two types of occlusion, namely covering the flame with a black or red rectangle for non-fire objects and fire object, respectively. Table 3 shows a comparison between the fire detection results of each network against the 6 fire images in Figure 10. The proposed MobileNet-Fire network correctly classified both fire and non-fire objects in the 6 fire images and 12 occluded images. In contrast, the existing CNN methods generated false alarms, especially for occluded images. This experiment shows that MobileNet-Fire has robust performance even when occlusion occurs. Figure 11 shows the visualization of the flame region in the image using Grad-CAM [49] by showing the activation of the last convolutional layer for the fire class.

V. CONCLUSION

We proposed the MobileNet-Fire network, which has a lower computational cost and achieves better performance compared with existing CNNs. Depthwise separable convolution, which is the basis of MobileNet-Fire, learns weights channel-wise. When classifying objects having specific colors, the channel multiplier can enhance the effect of depthwise separable convolution by emphasizing specific color channels. In addition, the SE block improves the channel-wise global representation of the results of depthwise separable convolution, allowing more effective extraction of the features of an object. Furthermore, the channel multiplier can be easily applied to networks to classify objects of specific color, other than fire.

We used Swish10, which is a custom Swish, as the activation function. Swish10 clips large weights and preserves the weights that are increased by the channel multiplier. The number of layers and feature maps were determined experimentally so that the features suitable for fire objects could be extracted effectively, thereby reducing the number of parameters. MobileNet-Fire achieved 95.44% accuracy for fire detection, which is higher than those of the existing networks compared.

Although our proposed algorithm focuses on fire detection, its detection performance is also better than those of the existing algorithms considered in this study, for non-fire objects with specific colors.

REFERENCES

[1] B. C. Ko, K. H. Cheong, and J. Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Saf. J.*, vol. 44, no. 3, pp. 322–329, Apr. 2009.

[2] P. V. K. Borges and E. Izquierdo, "A Probabilistic approach for vision-based fire detection in videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 5, pp. 721–731, May 2010.

[3] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 3, Oct. 2004, pp. 1707–1710.

[4] Z. Li, L. S. Mihaylova, O. Isupova, and L. Rossi, "Autonomous flame detection in videos with a Dirichlet process Gaussian mixture color model," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1146–1154, 2017.

[5] T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu, "Fire detection using statistical color model in video sequences," *J. Vis. Commun. Image Represent.*, vol. 18, no. 2, pp. 176–185, Apr. 2007.

[6] J.-Y. Kuo, T.-Y. Lai, Y.-Y. Fanjiang, F.-C. Huang, and Y.-H. Liao, "A behavior-based flame detection method for a real-time video surveillance system," *J. Chin. Inst. Eng.*, vol. 38, no. 7, pp. 947–958, 2015.

[7] B. C. Ko, K. H. Cheong, and J. Y. Nam, "Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian networks," *Fire Saf. J.*, vol. 45, no. 4, pp. 262–270, Jun. 2010.

[8] Y. H. Habiboğlu and O. Günay, "Covariance matrix-based fire and flame detection method in video," *Mach. Vis. Appl.*, vol. 23, no. 6, pp. 1103–1113, Nov. 2012.

[9] 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.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[13] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," May 2015, *arXiv:1505.00387*. [Online]. Available: https://arxiv.org/abs/1505.00387

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Feb. 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 4278–4284.

[19] S. Xie, R. Girshick, and P. Dollàr, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1492–1500.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[23] K. He, G. Gkioxari, and P. Dollàr, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[24] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1251–1258.

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[27] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[28] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 116–131.

[29] Z. Zhong, M. Wang, Y. Shi, and W. Gao, "A convolutional neural network-based flame detection method in video sequence," *Signal, Image Video Process.*, vol. 12, no. 8, pp. 1619–1627, Nov. 2018.

[30] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *Proc. IECON 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2016, pp. 877–882.

[31] A. J. Dunnings and T. P. Breckon, "Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1558–1562.

[32] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.

[33] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3113–3122, May 2019.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[35] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[37] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," May 2015, *arXiv:1505.00853*. [Online]. Available: https://arxiv.org/abs/1505.00853

[38] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," Oct. 2017, *arXiv:1710.05941*. [Online]. Available: https://arxiv.org/abs/1710.05941

[39] A. Chenebert, T. P. Breckon, and A. Gaszczak, "A non-temporal texture driven approach to real-time fire detection," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1741–1744.

[40] V. Hüttner, C. R. Steffens, and S. S. da Costa Botelho, "First response fire combat: Deep leaning based visible fire detection," in *Proc. Latin Amer. Robot. Symp. (LARS) Brazilian Symp. Robot. (SBR)*, Nov. 2017, pp. 1–6.

[41] C. R. Steffens, S. S. D. C. Botelho, and R. N. Rodrigues, "A texture driven approach for visible spectrum fire detection on mobile robots," in *Proc. Latin Amer. Robot. Symp. IV Brazilian Robot. Symp. (LARS/SBR)*, Oct. 2016, pp. 257–262.

[42] C. R. Steffens, R. N. Rodrigues, and S. S. da Costa Botelho, "An unconstrained dataset for non-stationary video based fire detection," in *Proc. Latin Amer. Robot. Symp. IV Brazilian Robot. Symp. (LARS/SBR)*, Oct. 2015, pp. 25–30.

[43] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999.

[44] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o (1/k^ 2)," in *Proc. Doklady AN USSR*, vol. 269, 1983, pp. 543–547.

[45] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[46] M. D. Zeiler, "Adadelta: An adaptive learning rate method," Dec. 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

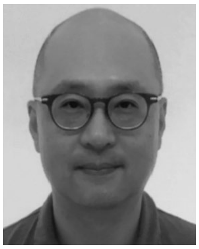
[48] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4148–4158.

[49] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 618–626.

HUNJUN YANG received the B.S. degree in electronic engineering and the M.S. degree in information engineering from Inha University, South Korea, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree in electronic engineering. His research interest includes deep learning and computer vision.

HYEOK JANG received the B.S. degree in electronic engineering, the M.S. degree in information engineering, and the Ph.D. degree in electronic engineering from Inha University, South Korea, in 1997, 1999, and 2015, respectively. From 2000 to 2001, he was a Researcher with the X-ray Systems Laboratory, Samsung Medison. From 2002 to 2009, he was a Senior Researcher with the Laboratory of INFINITT Healthcare. Since 2010, he has been a Senior Member of Research Staff with the Electronics and Telecommunications Research Institute (ETRI), South Korea. His current research interests include machine vision, artificial intelligence, and video surveillance.

TAEYONG KIM received the B.S. degree in electronic engineering from Inha University, South Korea, in 2016, and the M.S. degree in electrical engineering from McGill University, Canada, in 2019. He is currently with Hyundai Robotics, South Korea. His research interest includes the development of human–computer interaction and machine learning.



BOWON LEE (S'00–M'07–SM'12) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign, in 2003 and 2006, respectively. From 2007 to 2014, he was a Research Scientist with the Hewlett-Packard Laboratories in Palo Alto, California. He joined the faculty of the Department of Electronic Engineering, Inha University, in March 2014. His research interests include machine learning for signal processing, audio and speech signal processing, microphone array signal processing, acoustic event detection and localization, and multimodal signal processing and analysis. He is a member of the Audio Engineering Society. He has served as Awards Chair for ICASSP 2018 and in the technical program committee of numerous IEEE conferences and workshops.