IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 June 2023, accepted 4 August 2023, date of publication 15 August 2023, date of current version 21 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3305455

RESEARCH ARTICLE

Lightweight Convolutional Neural Network for Fire Classification in Surveillance System

DUY-LINH NGUYEN[1], (Member, IEEE), MUHAMAD DWISNANTO PUTRO[2], (Member, IEEE), AND KANG-HYUN JO[1], (Senior Member, IEEE)

[1]Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea
[2]Department of Electrical Engineering, Sam Ratulangi Unversity, Manado 95115, Indonesia

Corresponding author: Kang-Hyun Jo (acejo@ulsan.ac.kr)

This work was supported by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003.

ABSTRACT Fire is one of the worst disasters for human life. Fire can happen anywhere and the leading cause can be natural or man. Over the last century, scientists have invented sensor-based methods to minimize damage and provide early warning of fires. However, these applications are only applied in a limited space and distance. For the purpose of fire remote warning and deploying on low-computing devices, this paper proposes a vision-based method using a lightweight convolutional neural network architecture combined with the inception and attention mechanisms. This proposed network includes two main modules: a feature extractor and a classifier. The feature extractor exploits convolution layers, depthwise separable convolution layers, inception module, and attention mechanism to extract high-level feature maps. Next, the classifier applies the global average pooling layer to quickly reduce the feature map dimensions and uses the softmax function to calculate the probability of each class. The experiments performed the training and evaluation on six datasets with an accuracy of over 96%. The fire surveillance system was implemented with simulation videos on GPU, CPU, and Jetson Nano devices, with the highest speeds of 200.95 FPS, 31.08 FPS, and 14.27 FPS, respectively. A set of demonstration videos, source code, and proposed dataset are provided here: https://bit.ly/3Wlpycf.

INDEX TERMS Convolutional neural network (CNN), fire classification, fire surveillance system, inception module, squeeze and excitation attention module.

I. INTRODUCTION

Disasters caused by fire are still one of the major threats to humans [1]. According to an analysis from The Center for Research on the Epidemiology of Disasters (CRED), there have been 19 severe wildfires worldwide, claiming the lives of 90 people, and causing about 3.3 million USD of damage in the year 2021 [2]. Also in that year, another report [3] told that forest fires in Europe occurred in 39 countries with 1,113,464 hectares damaged. In addition, there are many large and small fires that are occurring every day in factories, companies, apartment buildings, amusement parks, and people's houses. The main causes of fires are natural factors such as thunder, earthquakes, volcanoes, and global warming, or carelessness in people's activities such as burning forests for farming, burning garbage, short-circuiting electric devices, and traffic accidents. This above observation shows that fires can appear easily and it has extreme effects on human lives and economic losses. In the last centuries, researchers have focused on developing fire warning devices to reduce the risks. These methods are mainly based on the performance and operation of sensors like heat sensors, smoke sensors, and flame sensors [4]. The sensor-based methods are usually simple to install but they can give false alarms due to their situational discrimination in narrow spaces. Later, the vision-based methods were widely applied to overcome the above disadvantages [5] and used in combination with sensor-based methods to improve fire warning efficiency in intelligent systems with larger warning spaces [6]. Following the development trend of vision-based methods and the CNN

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Da Lin.

application on low-computing devices, this paper proposes a fire surveillance system based on a lightweight convolutional neural network (CNN). The proposed network is a combination of convolution (Conv) layers, depthwise separable convolution (DWConv) layers, an inception module, and a squeeze and excitation attention mechanism (SE) in a feature extractor and it is finished with a classifier using a global average pooling layer and a softmax function. To diversify the context for fire classification, this work also proposes a fire classification dataset consisting of fire thermal images collected from various data sources. The entire network is trained and evaluated on six fire datasets (five available fire datasets and one proposed dataset). In addition, this work also conducts testing with simulated fire videos operating in real-time without high latency. The main contributions of this paper are shown as follows:

1) Proposes a lightweight and efficient CNN network architecture for fire classification. It consists of two main modules, a feature extractor and a classifier.
2) Exploits the advantages of the DWConv layers in the baseline stem and inception module design, replacing all fully connected layers in the traditional classifier with just one global average pooling layer to minimize network parameters.
3) Proposes a thermal imaging dataset for fire classification with all fire thermal images, named ThermalFire.
4) Deploys a fire surveillance system on low-computing devices such as CPU and embedded devices with high accuracy and negligible latency. This system can be applied to both indoor and outdoor environments.

The rest of the paper is organized as follows: Section II reviews the related approaches to fire classification and their advantages and disadvantages. Section III describes the proposed method in detail. Section IV analyzes the experiments and results. The last section contains the conclusion of the paper and future work.

II. RELATED WORKS

A. TRADITIONAL METHODS

The traditional fire detection methods are mainly based on the analysis of fire color features, fire motions and shapes, and combined techniques. Fire color characteristics were distinguished through the primary color channels such as RGB [7], YCbCr [8], YUV [9], and YUC [10], [11]. These methods were simple to implement and can achieve fire classification accuracy of 71.43% to 98.89%, but they required careful preprocessing to enhance relevant features and reduce noise. Regarding fire motions and shapes, the wavelet analysis and disorder characteristics method was used in [12] to detect fire and smoke. The combination of both analytic methods helped the fire and smoke detection system to optimize the minimum alarm ability for the indoor and outdoor systems. Other work in [13] proposed a method based on the Lucas-Kanade optical flow algorithm for building a fire detection system in the monocular video. Experimental results showed that this approach could achieve accuracy from 74.19% to 100% on monocular videos. To overcome several disadvantages of the optical flow algorithm, [14] designed two novel optical flow estimators: Optimal Mass Transport (OMT) and Non-Smooth Data (NSD) which are used for fire detection. The proposed method was evaluated on a large video database to demonstrate its superiority over related methods. In addition, studies in [15] and [16] combined fire color information and fire motion and shape analysis to improve fire detection accuracy. The experimental results achieved accuracies of 92.59% and 99.65% on the collected datasets, respectively. In general, traditional methods were easy to implement with low-computation devices and common sensors. However, accuracy in detecting large or tiny flames was a big challenge. In addition, they also required careful manual feature extraction.

B. CNN-BASED METHODS

In the computer vision field, fire detection has also been studied with many different CNN network architectures. Exploiting basic classification network architectures, [17] used pre-trained VGG16, ResNet50, and fine-tuning based on a fully connected layer to detect fire with a small dataset and low accuracy. Other classification network architectures such as AlexNet, SqueezeNet, and GoogleNet are also tuned for fire detection in surveillance systems [18], [19], [20]. These three techniques had quite high accuracy (from 94.3% to 94.5%), but they contained a large number of network parameters. The work in [21] proposed a fire detection network named Fire Detection for Jetson Nano and compares the performance with AlexNet and SqueezeNet architectures. This proposed network achieved the best accuracy of 84% and 79.66% on two datasets with very small model scale. Reference [22] designed a lightweight and efficient octave convolutional neural network for fire recognition in visual scenes. This study evaluated the performance by using cross-dataset validation and obtained an accuracy from 77.88% to 100%. Based on an analysis of experimental results of popular classification networks, [23] proposed an efficient CNN for fire detection in uncertain surveillance scenarios. This work conducted experiments with bigger fire datasets with quite high accuracy up to 100% but still maintains a large number of network parameters. These methods limited the deployment on low-computing devices for real-time system applications. Besides, the fire thermal imaging dataset is limited and unavailable, and its applications are not yet widely deployed [24]. These are major obstacles to the development of real-time fire detection and warning applications on low-computing devices in CNN-based methods.

III. PROPOSED METHODOLOGY

The proposed network is designed based on two main modules: a feature extractor and a classifier. The overall proposed network is shown in Figure 1.

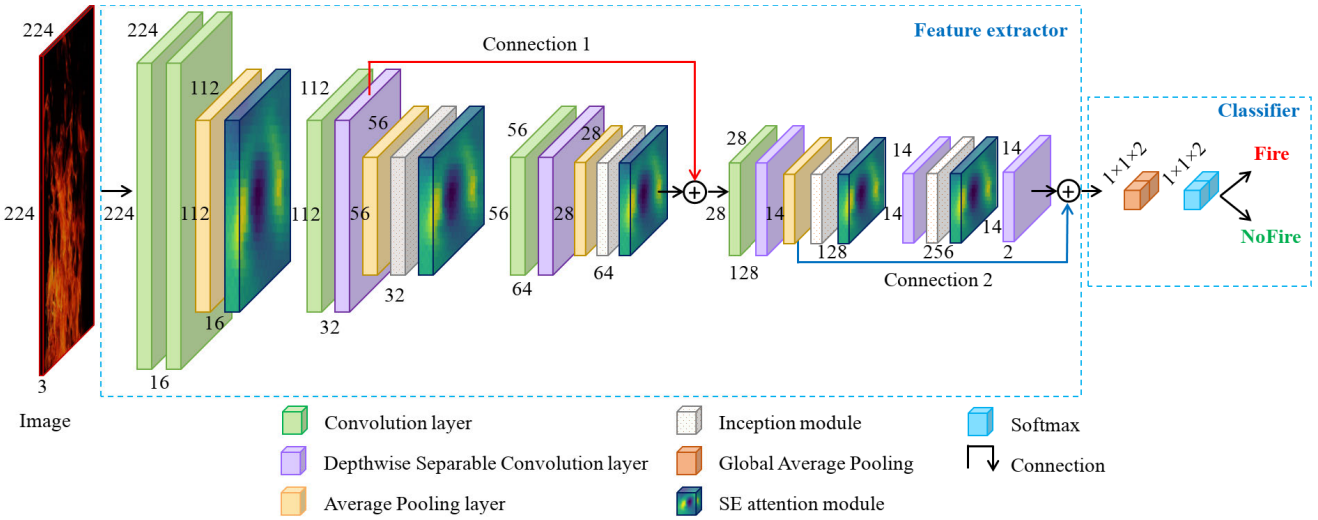

FIGURE 1. Architecture of the proposed fire classification network.

A. FEATURE EXTRACTOR

The feature extractor is a combination of Conv layers, DWConv layers [25], average pooling layers, inception module [26], SE attention modules [27], and proposed connection modules for high-level feature extraction. More specifically, this module consists of five main blocks. These blocks have the same structure except for the first and last blocks. In the first block, two 7 × 7 Conv layers are sequentially arranged followed by batch normalization (BN) and rectified linear unit (ReLU) activation function. The next ones are an average pooling layer and an SE attention module. In this case, Conv layers are used with a large kernel size to increase the receptive field to capture the basic information of the objects in the input image. Besides, the SE attention module guides the network to learn salient features at low-level feature maps. The original image size will be halved from 224 × 224 to 112 × 112 with 32 channels when passing through this first block.

The intermediate stage is organized with three identical blocks. Each block contains a 3 × 3 Conv layer, a 3 × 3 DWConv layer, an average pooling layer, an inception module, and an SE attention module. The feature map that goes through each block is reduced in size by two times and the number of channels is also increased by two times. Specifically, 112 × 112 × 32 feature map will become 14 × 14 × 128 feature map in these stages. The final block in the feature extractor is the stack of a DWConv layer, an inception module, an SE attention module, and then another DWConv layer. This block further enriches the informative feature map and guides the network to capture the salient features at a high level. After going through the final block, the feature map dimension remains 14 × 14 but the number of channels is reduced to two corresponding to the two class labels in the dataset.

Besides mentioned layers, the connections play a very important role in the feature extractor. They merge the previous feature map and the current feature map by an addition operator to generate the output feature map. This technique enhances the useful information in the feature map at different feature levels. Combining ideas from Residual [28] and Inception [26] networks, this paper proposes two versions of the connection as shown in Figure 2. Each connection consists of two branches: the main branch and the sub-branch. The sub-branch in connection 1 uses a 3 × 3 DWConv layer, an average pooling layer, and a 1 × 1 DWConv layer followed by a BN layer. The sub-branch in connection 2 has a similar structure to connection 1 without an average pooling layer. Both connections perform the task of extracting the low-level feature map and merging it with the current feature map via the addition operator. This technique aims to mix information between different feature map levels to further increase useful information for feature extraction. In this work, connection 1 links the 1^{st} and 3^{rd} blocks while connecting 2 links the 4^{th} and last block. The calculation formula of the connection is described as follows:

$$y_i = \mathcal{I}_i(x) + \mathcal{I}'_i(x), \tag{1}$$

where i is the version of connection ($i = 1$ or 2) and x and y are the input and output feature maps, respectively. $\mathcal{I}_i(x) \in \mathbb{R}^{W\times H\times C}$ is the output feature map of the main branch. $\mathcal{I}'_i(x) \in \mathbb{R}^{W\times H\times C}$ is the output feature map of the sub-branch. Details for each sub-branch version are as follows:

$$\begin{aligned}\mathcal{I}'_1(x) &= BN(DWConv^{1\times1}(\mathcal{A}^{3\times3}(DWConv^{3\times3}(x)))),\\ \mathcal{I}'_2(x) &= BN(DWConv^{1\times1}(DWConv^{3\times3}(x))),\end{aligned} \tag{2}$$

where BN is the batch normalization layer. $DWConv^{1\times1}$ and $DWConv^{3\times3}$ are 1 × 1 and 3 × 3 depthwise separable convolution layers, respectively. $\mathcal{A}^{3\times3}$ is the 3 × 3 average pooling layer.

The inception modules are designed based on the original architecture in [25] with the complete replacement of the

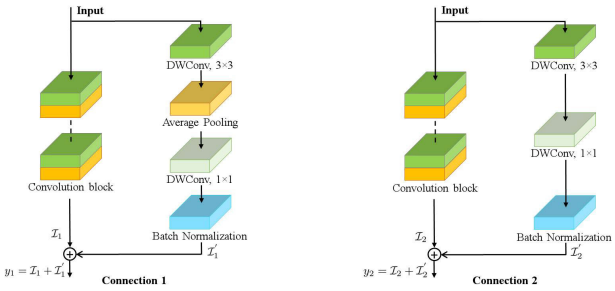

FIGURE 2. Architecture of the connection modules. The connections help to merge the feature maps at different levels but insignificantly increase the network parameters. On the other hand, the use of different architectures also aims to adapt to each feature map level.

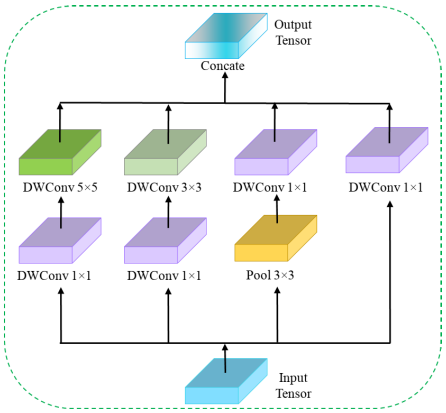

FIGURE 3. Architecture of the inception module.

Conv layers by the DWConv layers. The structure of each inception module is shown in Figure 3 which consists of four branches with different numbers and types of layers. From left to right, the first branch is a combination of a 1×1 DWConv layer and a 5×5 DWConv layer, the second is a combination of a 1×1 DWConv layer and a 3×3 DWConv layer, the third is a combination of a 3×3 max pooling layer and 1×1 DWConv layer, and the final branch uses only one 1×1 DWConv layer. The output feature maps from four branches are then combined using the concatenation operation to create the refined feature map. The combined use of DWConv layers with different kernel sizes and an average pooling layer allows the inception module to capture information across multiple receptive fields. Therefore, these modules enrich the feature map information before moving on to the next module. In addition, the replacement of the Conv layers by the DWConv layers has significantly reduced the network parameters but still ensures information for the feature extraction process.

The SE attention module is composed of three operations: squeeze (F_{sq}), excitation (F_{ex}), and scaling (F_{scale}). The architecture of the SE attention module is presented in Figure 4. Assume the input feature map is $X \in \mathbb{R}^{H' \times W' \times C'}$ with dimensions H' height, W' width, and C' channels. Apply the transformation F_{tr} (such as convolutional operator) to

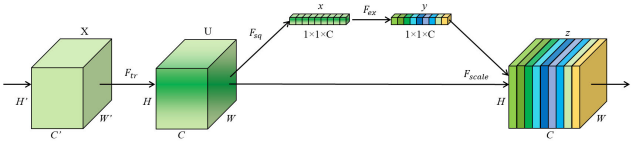

FIGURE 4. Architecture of the squeeze-and-excitation attention module. Different colors in the feature maps depict different attention levels per channel.

obtain the feature map $U \in \mathbb{R}^{H \times W \times C}$. In the squeeze operator, a global average pooling is used to generate channel-wise attention. To do that, it shrinks the feature map U through its $H \times W$ spatial dimensions. Therefore, each c channel of the feature map U is calculated as follows:

$$x_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j), \tag{3}$$

The excitation operator aims to capture channel-wise dependencies. This operator gets the output of the squeeze stage to generate an activation vector. The process is calculated based on two fully connected layers with a bottleneck with ratio r (ratio of the network node of the current layer to the input layer) and a sigmoid activation function. The formula for calculating the y activation vector is:

$$y = F_{ex}(x, W) = \sigma(W_2 ReLU(W_1 x), \tag{4}$$

where σ is the sigmoid activation function, $ReLU$ is the rectified linear activation function, $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$, $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$, and $x = [x_1, x_2, \ldots, x_c]$.

Finally, the scale operator modifies the U feature map and the y activation vector by the channel-wise multiplication operator ($\otimes$). The calculation formula is shown below:

$$z_c = F_{scale}(u_c, y_c) = u_c \otimes y_c. \tag{5}$$

The stack of refined feature maps z_c to form an output feature map $z = [z_1, z_2, \ldots, z_c]$ which is the same dimension as the original feature map U.

B. CLASSIFIER

In common classification network architectures, the fully connected layers are used at the end of the feature extractor. With this technique, the number of connections between network nodes increases significantly. It is also the agent that increases the amount of computation on the network that hinders implementation in real-time systems. To solve this problem, this paper proposes to replace all fully connected layers with only one average pooling layer. Therefore, the spatial features are extracted per channel using the average operation. Specifically, from the last feature map of the feature extractor with a $14 \times 14 \times 2$ dimension, it will quickly reduce to $1 \times 1 \times 2$. Then, a softmax function is applied to calculate the probability of each class (*Fire* and *NoFire*).

TABLE 1. Datasets and the ratio of training and evaluation sets.

Dataset	Fire	NoFire	Total	Train/Eval
FireNet	1,124	1,301	2,425	70%/30%
FireSense	329	577	906	80%/20%
CairFire	110	541	651	549/110
FireSmoke	1,000	1,000	2,000	1,800/200
FireDetection	8,847	21,703	30,550	8,032/22,518
ThermalFire	1,535	1,669	3,204	70%/30%

C. LOSS FUNCTION

During training, this work uses a categorical cross-entropy loss function [29] to evaluate the difference between the predicted value and the target value. This function is applied to two classes and is defined as follows:

$$L_{cls} = -\sum_{i=0}^{1} \mathcal{P}_i^* log(\mathcal{P}_i), \tag{6}$$

where i represents the index of each class in the dataset (0 to 1). $\mathcal{P}_i^*$ is the target indicator (0 or 1). $\mathcal{P}_i$ denotes the prediction probability from the network and log is a natural logarithm function.

D. FIRE VIDEO TESTING SYSTEM

The overall fire video testing system is described in detail in Figure 5 with the training phase and the testing phase. Focusing on the testing phase, it includes the input, classification process, and output. The input is a set of videos with different resolutions including VGA, HD, FHD, and thermal video. The classification is performed by the trained model on the FireNet dataset. The output is the signals displayed on the screen consisting of the class to be classified, the accuracy of that class, and the speed measured in frames per second (FPS). In fact, the system can replace the input with a camera and the output can integrate an additional speaker to broadcast an alarm when a fire is detected. This is also the structure of the real-time fire warning system.

IV. EXPERIMENTS

A. DATASETS

This experiment trains and evaluates the proposed fire classification network on six datasets: FireNet [6], FireSense [16], CairFire [17], FireSmoke [30], FireDetection [23], and proposed ThermalFire dataset. To enrich the training set and avoid overfitting issues, this experiment applies several data augmentation techniques: random zoom, random brightness, and random shift. The details of each dataset are described in Table 1.

1) FireNet DATASET

FireNet was a diverse dataset captured from fire and non-fire videos in a challenging environment and collected from various internet sources (Google and Flickr). To increase the richness of the dataset, the authors also applied data augmented techniques in the image sets. The result was a dataset containing 1,124 fire images and 1,301 non-fire images. To compare fairly with other experiments, this paper divides the FireNet dataset into 70% for the training set and 30% for the evaluation set.

2) FireSense DATASET

FireSense dataset was a video dataset containing twenty-seven videos for fire detection and twenty-two videos for smoke detection. The video set for fire detection has eleven videos with the appearance of fire and sixteen videos without fire. This experiment extracts frames from fire detection videos with 329 fire frames and 577 non-fire frames. All images are separated into 80% for the training set and 20% for the evaluation set.

3) CairFire DATASET

The CairFire dataset consists of 110 fire images and 541 non-fire images. These images were collected from the internet with many different fire situations, indoor and outdoor environments, and different lighting conditions similar to fire color. This experiment selects 541 images for training and 110 images for evaluation.

4) FireSmoke DATASET

The FireSmoke dataset contains 3,000 images divided into three classes: fire, neutral, and smoke. Each class consists of 1,000 images including 900 images for training and 100 images for evaluation. In this work, only images from two classes, fire and neutral (corresponding to ''no fire'' class) are used to train and evaluate the proposed network.

5) FireDetection DATASET

The FireDetection dataset was created mainly from two other datasets consisting of two classes, fire and non-fire, with 30,776 images. Following the settings in [23], this experiment selects 8,032 images for training and 22,518 images to evaluate the proposed network.

6) ThermalFire DATASET

The ThermalFire dataset is proposed by the authors in this paper. This dataset is built based on fire thermal videos [31], [32] and other thermal images collected from the internet [33], [34]. It contains 3,204 fire thermal images with 1,535 fire images and 1,669 no-fire images. This work divides the dataset into 70% for the training phase and 30% for the evaluation phase.

B. EXPERIMENTAL SETTING

The proposed fire classification network is built in the Python programming language with the Keras framework. This network is trained and evaluated on a Tesla-V100 GPU. Another GPU (GeForce GTX 1080Ti, 32GB of RAM), one CPU (Intel Core I7-4770 CPU @ 3.40 GHz, 32GB of RAM) and one Jetson Nano (Nvidia Maxwell GPU, 4GB of RAM) are used to test the real-time video system with different resolutions: VGA (640 × 480 pixels), HD (1280 × 720 pixels), and FHD

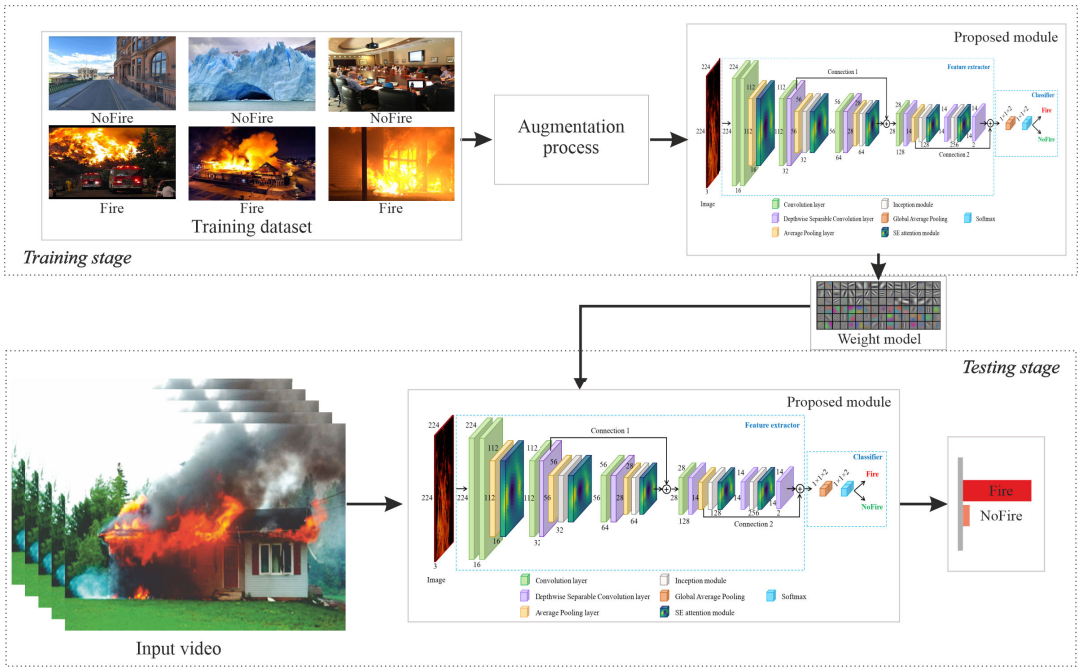

FIGURE 5. Fire video testing system.

TABLE 2. The comparison accuracy (%) with different methods on six datasets. The red colored numbers represent the best competitors. The † symbol denotes the finetuned classification networks. The N/A symbol denotes the no available values.

Model/Method	Parameters	GFLOPs	FireNet	FireSense	CairFire	FireSmoke	FireDetection	ThermalFire
Proposed	**579,986**	2.981000	**96.08**	**100**	**97.27**	**98.50**	**100**	**100**
SqueezeNet†	258,874	0.438500	93.14	100	83.06	98.05	99.99	90.48
MobilleNetV2†	3,571,778	0.002627	94.26	100	91.99	97.83	95.36	94.93
MobilleNetV1†	4,288,714	0.002102	94.96	95.14	94.17	96.50	99.89	100
VGG13†	5,220,418	0.001054	92.58	87.03	83.61	97.00	100	100
NASNetMobile†	5,362,334	1.148000	92.58	96.76	92.17	95.00	98.75	99.47
DenseNet†	8,097,354	0.002102	95.24	100	93.08	98.50	97.27	99.68
VGG16†	15,250,250	0.001054	95.66	100	92.71	96.50	100	99.89
VGG19†	20,559,946	0.001054	96.08	100	93.44	98.00	100	99.89
Xception†	22,969,906	0.001054	95.66	100	92.17	97.00	100	100
InceptionV3†	23,911,210	0.004199	93.00	96.76	94.17	95.00	99.56	99.47
LeNet†	78,432,080	0.937800	89.64	100	87.43	92.50	100	100
AlexNet [21]	58,650,560	N/A	N/A	N/A	78.50	81.66	N/A	N/A
SqueezeNet [21]	421,098	N/A	N/A	N/A	87.00	77.33	N/A	N/A
Fire Detection [21]	7,380	N/A	N/A	N/A	84.00	79.66	N/A	N/A
CNN in [22]	956,226	N/A	85.14	100	90.78	N/A	N/A	N/A
EMN_Fire [23]	N/A	N/A	N/A	N/A	N/A	N/A	95.86	N/A
CNNFire [19]	N/A	N/A	N/A	N/A	N/A	N/A	94.61	N/A
GNetFire [20]	N/A	N/A	N/A	N/A	N/A	N/A	93.66	N/A
ANetFire [18]	N/A	N/A	N/A	N/A	N/A	N/A	94.27	N/A
MES [10]	N/A	N/A	N/A	N/A	N/A	N/A	93.55	N/A
Method in [14]	N/A	N/A	N/A	N/A	N/A	N/A	92.86	N/A
Method in [11]	N/A	N/A	N/A	N/A	N/A	N/A	90.32	N/A
RGB [12]	N/A	N/A	N/A	N/A	N/A	N/A	74.20	N/A
YUV [12]	N/A	N/A	N/A	N/A	N/A	N/A	87.10	N/A
Method in [8]	N/A	N/A	N/A	N/A	N/A	N/A	83.87	N/A
Method in [7]	N/A	N/A	N/A	N/A	N/A	N/A	87.10	N/A

(1920 × 1080 pixels) and thermal video. The training process goes through 300 epochs with a batch size of 32. The learning rate is initialized at 10^{-3} and decremented by a factor of 0.65 times after 10 epochs if the accuracy is not improved. The Adam optimization method [35] is applied to update the weight during training. Several data augmentation methods are used such as random clipping, rotation, and flipping to contribute to reducing overfitting.

C. RESULT ANALYSIS

The proposed fire classifier network is trained and evaluated on the six datasets described in Section IV-A. On the other

FIGURE 6. The qualitative fire classification on six datasets.

hand, it is also tested with different video resolutions (VGA, HD, FHD) and thermal video on one GPU, one CPU, and one Jetson Nano device. The results are reported through accuracy (%) and network parameters. As a result, the proposed network achieved accuracies of 96.08%, 100%, 97.27%, 98.50%, 100%, and 100% on the FireNet, FireSense, CairFire, Firesmoke, FireDetection, and ThermalFire datasets respectively. In common, the network parameter is the sum of the weight and bias of Conv, DWConv, and fully connected layers used in the network. This study restricts the use of the Conv layers, encourages the use of the DWConv layer both in the baseline and other modules, and replaces the entire fully connected layer with a single global average pooling layer. This technique helps to optimize network parameters significantly. Therefore, the whole proposed network contains only 579,986 parameters but is still computationally complex enough with 2,981 GFLOPS to ensure feature extraction at different levels. To evaluate the performance of the proposed network, the experiment is compared with other network architectures in the same datasets. In addition, this work also refined, retrained and evaluated the typical classification network architectures across all six datasets. The comparison of the proposed network and different methods on six datasets are shown in Table 2. For the FireNet dataset, the proposed network outperforms the refined networks and the CNN in [22]. Specifically, it achieves an accuracy equal to the best competitor (VGG19) of 96.08% but its network parameter is 35.45 times less. For the FireSense dataset, the proposed network achieves absolute accuracy along with the refined networks (SqueezeNet, MobilleNetV2, DenseNet, VGG16, VGG19, Xception, and LeNet) and the CNN in [22] and outperforms other methods. Compared to the best competitor

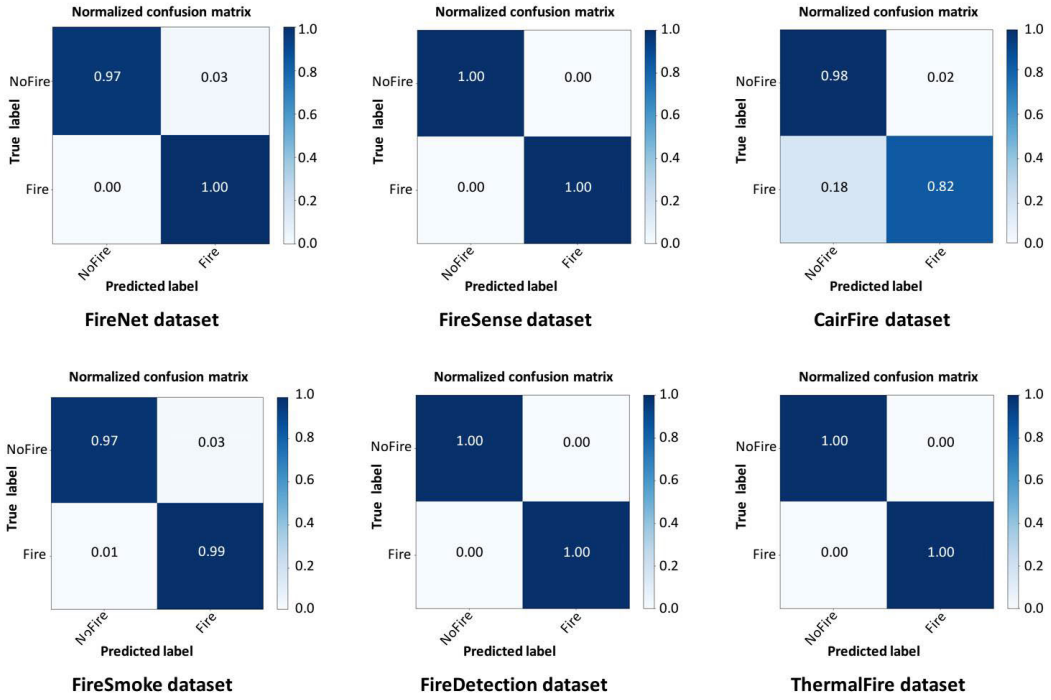

FIGURE 7. The confusion matrices on six datasets.

FIGURE 8. The GRAD-CAM visualization on the FireSmoke dataset. The above images are the original images and the below ones are GRAD-CAM visualization images.

(SqueezeNet), it has more than twice the network parameters. For the CairFire dataset, the proposed network also outperforms the other refined networks and classification networks in [21] and [22]. It achieves an accuracy of 3.1% better than the best competitor (MobilleNetV1) while its network parameter is 7.39 times less. Similar to the FireNet dataset, the accuracy of the proposed network tends to be similar. It also outperforms the refined networks and the CNN proposed in [21]. When compared with the best competitor (DenseNet), it reaches the same accuracy, but the network parameter is 13.96 times less. The proposed network again achieves absolute accuracy in the FireDetection dataset with network architectures such as the VGG network family and the Xception network. In addition, it also outperformed the networks in [7], [8], [10], [11], [12], [14], [18], [19], [20], and [23] with an accuracy from 4.14% to 16.13% better. When compared with the best competitor (VGG13), its network parameter is 9.0 times less. For the ThermalFire dataset, this network also achieves absolute accuracy along with MobileNetV1, Exception, and LeNet networks. Compared with the best competitor (MobileNetV1), its network parameters have 7.39 times fewer. Figure 6 presents the qualitative results of the proposed network on six datasets. Figure 7 shows confusion matrices on six datasets and demonstrates

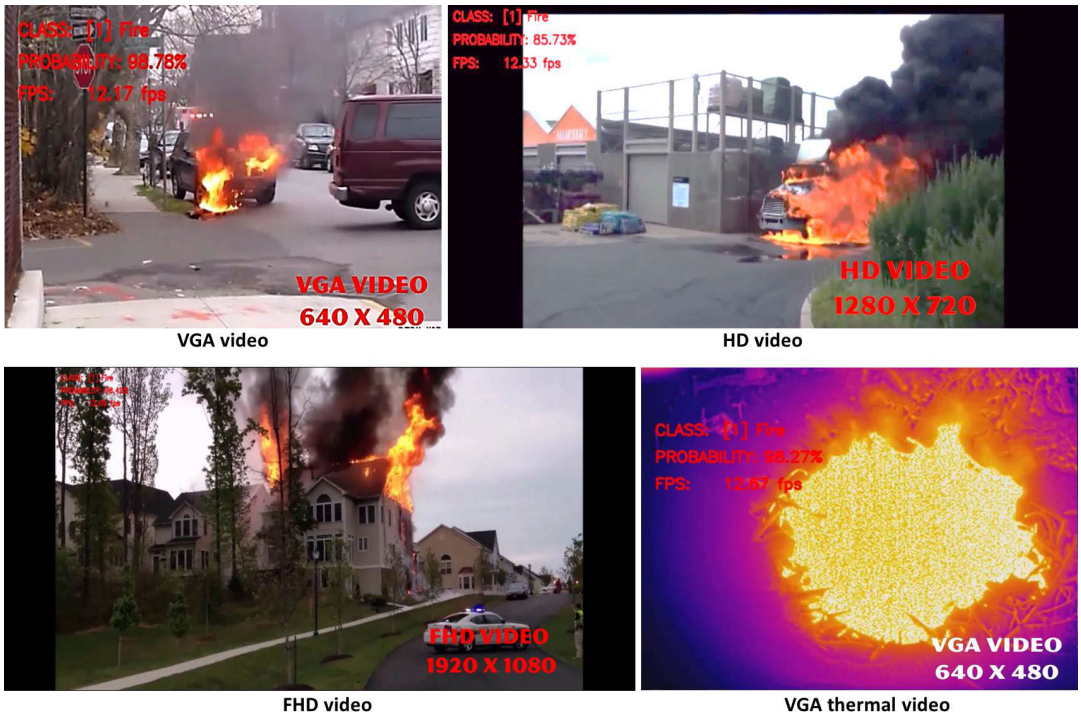

FIGURE 9. The results of the video testing system with VGA, HD, FHD, and VGA thermal video on Jetson Nano device (Nvidia Maxwell GPU, 4GB of RAM).

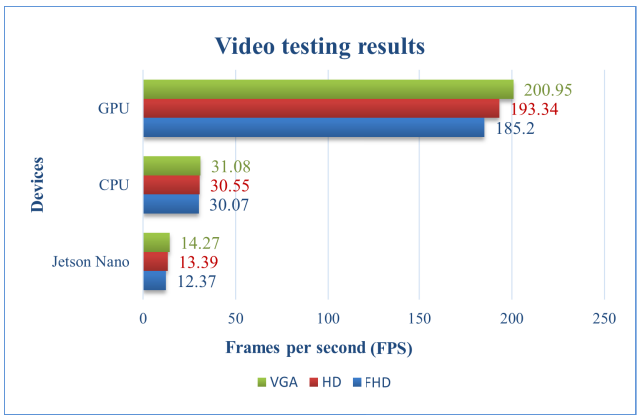

FIGURE 10. The speed of the video testing system with VGA, HD, and FHD on the FireNet dataset and three devices: GPU (GeForce GTX 1080Ti, 32GB of RAM), CPU (Intel Core I7-4770 CPU @ 3.40 GHz, 32GB of RAM), and Jetson Nano (Nvidia Maxwell GPU, 4GB of RAM).

that the proposed network can achieve balanced accuracy between two classes of *Fire* and *NoFire* (on FireNet, CairFire, and FireSmoke datasets) or absolute accuracy (on FireSense, FireDetection, and ThermalFire datasets). The proposed network is also capable to distinguish the fire and background feature in order to focus on learning fire-related properties as shown in Figure 8 based on the GRAD-CAM technique [36]. This functionality is obtained by the operating mechanism of the SE attention modules embedded in the feature extraction process. To test the speed of the proposed network in a real-time system, the experiment is conducted with a set of videos and a pre-trained model on the FireNet dataset as the system described in Section III-D. This system is subject to changing the videos with a camera to deploy experiments in real-time. For safety reasons, the experiment was performed only on a set of fire simulation videos recorded from actual fires. Several experimental results with videos on the Jetson Nano device are shown in Figure 9. The results from Figure 10 demonstrate that the proposed network achieves maximum speeds of 200.95 FPS, 31.08 FPS, and 14.27 FPS on the GPU, CPU, and Jetson Nano devices, respectively. This speed will gradually decrease according to the resolution from VGA to HD to FHD with a minimum speed of 185.2 FPS, 30.55 FPS, and 12.37 FPS, respectively. With this minimum speed, the system can perfectly work in real-time with low-computing devices and embedded devices. This encourages the use of VGA resolution video when implementing the system in practice because it ensures the alarm signal display requirements and the operating speed of the system. During the testing on video, it was shown that the proposed network works stably in small indoor spaces, but also reveals several disadvantages when deployed to monitor the outdoor environment. Sometimes the network cannot distinguish the fire color and lights similar to street lights, car lights, etc. Even small fires are difficult to detect because the fire shape can change continuously. Besides, the occlusion issue is also a challenge for the proposed system when detecting remote fire sources. The speed of camera movement also greatly affects the accuracy of the network. Therefore,

TABLE 3. Ablation study 1 on the FireNet dataset. The red number represents the best network.

Modules	Network					
Baseline	✓	✓	✓	✓	✓	✓
Inception		✓	✓			✓
Connection			✓	✓		✓
SE					✓	✓
#Parameters	215,768	409,176	570,720	367,232	238,150	579,986
GFLOPS	2.369	2.906	2.299	2.438	2.369	2.981
Accuracy (%)	95.46	69.33	95.52	94.26	95.80	96.08

establishing the appropriate scene range and placement camera angle that allows the network to achieve the required accuracy is a challenge to design a fire surveillance system.

D. ABLATION STUDIES

This research evaluates the impact of each proposed module on the entire network through four ablation studies. Ablation study 1 examines the influence of the Inception, Connection, and SE attention modules and their combinations. Ablation study 2 inspects the advantages of the SE attention module compared to other attention mechanisms such as BAM and CBAM. Ablation study 3 compares the efficiency of the GAP and fully connected (FC) layer. Ablation study 4 focuses on comparing Conv and DWConv layers when used to design the inception module.

1) ABLATION STUDY 1

In this ablation study, a baseline network is designed that also includes two main modules. However, the feature extractor is only based on the basic elements in a CNN network such as Conv, DWCon, and average pooling layers to extract features at different levels. The baseline network is trained and evaluated on the FireNet dataset for comparison. In the next step, other modules are also integrated into the baseline network to form a new CNN and then this work conducts training and evaluation. The results in Table 3 show that integrating just one or two modules into the baseline network only slightly increases the accuracy (from 0.06% to 0.34%) or even reduces the accuracy significantly by about 26.13% (the case of the inception module). The integration of all proposed modules into the baseline network helps the network achieve the highest accuracy of 96.08%. However, it is also a trade-off with increasing network parameters of 2.69 times and computational complexity of 0.612 GFLOPS when compared to the baseline network.

2) ABLATION STUDY 2

To choose the appropriate attention mechanism for the proposed network, this work also evaluates the influence of each module. Specifically, each attention module is integrated with the connection, and inception modules into the baseline network, and then proceed with training and evaluation. The results in Table 4 show that when using the BAM and CBAM, the network parameters and accuracy are similar. In contrast, when replacing them with the SE attention module, the network contains of nearly two thousand fewer network parameters, and the accuracy increases by more than 0.5% while the computational complexity is also reduced. This is the reason why the proposed network uses the SE attention module as the main attention mechanism.

TABLE 4. Ablation study 2 with different attention mechanisms on the FireNet dataset. The red colored number represents the best network.

Modules	Network		
Baseline	✓	✓	✓
Inception	✓	✓	✓
Connection	✓	✓	✓
BAM	✓		
CBAM		✓	
SE			✓
#Parameters	588,064	588,162	579,986
GFLOPS	2.983	2.987	2.981
Accuracy (%)	95.50	95.52	96.08

TABLE 5. The Ablation 3 study compares FC and GAP usage in a network on the FireNet dataset. The red-colored number denotes the best network, FC-128 is the fully connected layer with one hidden layer of 128 notes.

Module	#Parameters	GFLOPS	Accuracy (%)
FC-128	630,548	2.981	92.16
GAP	579,986	2.981	96.08

TABLE 6. Ablation study 4 with different inception modules on the FireNet dataset. The red-colored number represents the best network.

Inception module	#Parameters	GFLOPS	Accuracy (%)
Conv layer	673,106	3.133	95.39
DWConv layer	579,986	2.981	96.08

3) ABLATION STUDY 3

As mentioned above, this work completely replaces the fully connected layers in the traditional classification network with only one GAP layer. Table 5 shows the comparison results between using FC and GAP. In this case, the classification network uses only one hidden layer with 128 network notes. The results show that using GAP dramatically reduces the network parameter (50,562 parameters) and increases the classification accuracy by a large margin (3.92%) while the computational complexity is the same.

4) ABLATION STUDY 4

The final ablation study focuses on evaluating the architecture of the inception module used in the proposed network. As mentioned above, the inception module uses the original architecture but replaces all Conv layers with

DWConv layers to optimize network parameters. Table 6 shows that this replacement can save 93,120 parameters, 0.152 GFLOPS computational complexity, and increase the accuracy to 0.69%.

V. CONCLUSION

This paper proposes a lightweight convolutional neural network for fire classification. This network exploits the basic components of a CNN network such as Conv, DWCon, average pooling layers combined with proposed connections, and inception modules to extract feature maps. In addition, the integration of SE attention modules at the end of each block aims to guide the network to focus on learning salient features at each level. Finally, the classifier module performs context classification in two classes, *Fire* and *NoFire*. The experiments were conducted on the image datasets and real-time simulated videos. This work also designs a fire surveillance system that works on low-computing devices with high accuracy and negligible latency. In the future, the system will be combined with a fire detector to solve the problem of small-scale fire classification and at night conditions with infrared cameras. On the other hand, the system will be integrated to send messages to users via mobile networks to alert anytime and anywhere.

REFERENCES

[1] M. Cavallini, M. Papagni, and F. W. B. Preis, "Fire disasters in the twentieth century," *Ann. Burns Fire Disasters*, vol. 20, pp. 101–103, Jun. 2007.
[2] *2021 Disasters in Numbers*. Accessed: Apr. 20, 2023. [Online]. Available: https://cred.be/sites/default/files/2021_EMDAT_report.pdf
[3] J. San-Miguel-Ayanz et al., "Advance report on wildfires in Europe, Middle East and North Africa 2021," Publications Office Eur. Union, Luxembourg, Tech. Rep. EUR 31028 EN, JRC128678, 2022, doi: 10.2760/039729.
[4] F. Khan, Z. Xu, J. Sun, F. M. Khan, A. Ahmed, and Y. Zhao, "Recent advances in sensors for fire detection," *Sensors*, vol. 22, no. 9, p. 3310, Apr. 2022.
[5] A. Filonenko, D. C. Hernández, and K.-H. Jo, "Fast smoke detection for video surveillance using CUDA," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 725–733, Feb. 2018.
[6] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "FireNet: A specialized lightweight fire & smoke detection model for real-time IoT applications," 2019, *arXiv:1905.11922*.
[7] 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, 2004, pp. 1707–1710.
[8] T. Çelik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Saf. J.*, vol. 44, no. 2, pp. 147–158, Feb. 2009.
[9] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images," *Fire Saf. J.*, vol. 41, no. 4, pp. 285–289, Jun. 2006.
[10] P. Foggia, A. Saggese, and M. Vento, "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 9, pp. 1545–1556, Sep. 2015.
[11] Y. Habiboglu, O. Günay, and A. Cetin, "Covariance matrix-based fire and flame detection method in video," *Mach. Vis. Appl.*, vol. 23, pp. 1–11, Nov. 2011.
[12] A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, "Fire and smoke detection using wavelet analysis and disorder characteristics," in *Proc. 3rd Int. Conf. Comput. Res. Develop.*, vol. 3, Mar. 2011, pp. 262–265.
[13] S. Rinsurongkawong, M. Ekpanyapong, and M. N. Dailey, "Fire detection for early fire alarm based on optical flow video processing," in *Proc. 9th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, May 2012, pp. 1–4.
[14] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2786–2797, Jul. 2013.
[15] R. Di Lascio, A. Greco, A. Saggese, and M. Vento, "Improving fire detection reliability by a combination of videoanalytics," in *Image Analysis and Recognition*, A. Campilho and M. Kamel, Eds. Cham, Switzerland: Springer, 2014, pp. 477–484.
[16] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 339–351, Feb. 2015.
[17] J. Sharma, O.-C. Granmo, M. Goodwin, and J. Fidje, "Deep convolutional neural networks for fire detection in images," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, Aug. 2017, pp. 183–193.
[18] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," *Neurocomputing*, vol. 288, pp. 30–42, May 2018.
[19] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.
[20] 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.
[21] J. Gotthans, T. Gotthans, and R. Marsalek, "Deep convolutional neural network for fire detection," in *Proc. 30th Int. Conf. Radioelektronika (RADIOELEKTRONIKA)*, Apr. 2020, pp. 1–6.
[22] A. Ayala, E. Lima, B. Fernandes, B. L. D. Bezerra, and F. Cruz, "Lightweight and efficient octave convolutional neural network for fire recognition," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2019, pp. 1–6.
[23] 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.
[24] S. Li, Y. Wang, C. Feng, D. Zhang, H. Li, W. Huang, and L. Shi, "A thermal imaging flame-detection model for firefighting robot based on YOLOv4-F model," *Fire*, vol. 5, no. 5, p. 172, Oct. 2022.
[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.
[29] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, vol. 133. New York, NY, USA: Springer, 2004.
[30] *Fire Smoke Dataset*. Accessed: Jul. 20, 2022. [Online]. Available: https://github.com/DeepQuestAI/Fire-Smoke-Dataset
[31] *Pwhoazin—Youtube*. Accessed: Dec. 14, 2022. [Online]. Available: https://www.youtube.com/@PWHoazin
[32] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé, and E. Blasch, "Aerial imagery pile burn detection using deep learning: The FLAME dataset," *Comput. Netw.*, vol. 193, Jul. 2021, Art. no. 108001.
[33] Q. Ashfaq, U. Akram, and R. Zafar. *Thermal Image Dataset for Object Classification*. Accessed: Dec. 14, 2022. [Online]. Available: https://data.mendeley.com/datasets/btmrycjpbj/1
[34] J. Nelson. *Thermal Dogs and People Object Detection Dataset*. Accessed: Dec. 14, 2022. [Online]. Available: https://public.roboflow.com/object-detection/thermal-dogs-and-people
[35] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Dec. 2014, doi: 10.48550/arXiv.1412.6980.
[36] 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," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Oct. 2019.

DUY-LINH NGUYEN (Member, IEEE) received the B.Eng. degree in applied informatics from the Vinh University of Technology Education, Vietnam, in 2010, and the master's degree in computer science from The University of Da Nang, Vietnam, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, South Korea. After the B.Eng. degree, he joined the Information Technology and Electrical Engineering Department, Quang Binh University, Vietnam, as a Lecturer. He was with the Intelligent System Laboratory (ISLab), Department of Electrical, Electronic, and Computer Engineering, University of Ulsan. His current research interests include object detection and recognition in computer vision based on machine learning.

MUHAMAD DWISNANTO PUTRO (Member, IEEE) received the B.Eng. (S.T.) degree in electrical engineering from Sam Ratulangi University, Manado, Indonesia, in 2010, the M.Eng. degree from the Department of Electrical Engineering, Gadjah Mada University, Yogyakarta, Indonesia, in 2012, and the Ph.D. degree from the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, South Korea, in 2022. His current research interests include computer vision and deep learning, which focuses on robotic vision and perception.

KANG-HYUN JO (Senior Member, IEEE) received the Ph.D. degree in computer-controlled machinery from Osaka University, Osaka, Japan, in 1997. After a year of experience with ETRI as a Postdoctoral Research Fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea, where he is currently the Faculty Dean of the School of Electrical Engineering. His current research interests include computer vision, robotics, autonomous vehicles, and ambient intelligence. He was the Director or an AdCom Member of the Institute of Control, Robotics and Systems and the Society of Instrument and Control Engineers, the IEEE IES Technical Committee on Human Factors Chair, an AdCom Member, and the Secretary, until 2019. He has also been involved in organizing many international conferences, such as the International Workshop on Frontiers of Computer Vision, the International Conference on Intelligent Computation, the International Conference on Industrial Technology, the International Conference on Human System Interactions, and the Annual Conference of the IEEE Industrial Electronics Society. He is currently an Editorial Board Member for international journals, such as the *International Journal of Control, Automation and Systems* and *Transactions on Computational Collective Intelligence*.