Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.034475
Article

Tech Science Press

Efficient Deep Learning Framework for Fire Detection in Complex Surveillance Environment

Naqqash Dilshad[1], Taimoor Khan[2] and JaeSeung Song[1],[*]

[1]Department of Convergence Engineering for Intelligent Drone, Seoul, 05006, Korea
[2]Department of Computer Science, Islamia College Peshawar, Peshawar, 25120, Pakistan
*Corresponding Author: JaeSeung Song. Email: jssong@sejong.ac.kr
Received: 18 July 2022; Accepted: 30 September 2022

Abstract: To prevent economic, social, and ecological damage, fire detection and management at an early stage are significant yet challenging. Although computationally complex networks have been developed, attention has been largely focused on improving accuracy, rather than focusing on real-time fire detection. Hence, in this study, the authors present an efficient fire detection framework termed E-FireNet for real-time detection in a complex surveillance environment. The proposed model architecture is inspired by the VGG16 network, with significant modifications including the entire removal of Block-5 and tweaking of the convolutional layers of Block-4. This results in higher performance with a reduced number of parameters and inference time. Moreover, smaller convolutional kernels are utilized, which are particularly designed to obtain the optimal details from input images, with numerous channels to assist in feature discrimination. In E-FireNet, three steps are involved: preprocessing of collected data, detection of fires using the proposed technique, and, if there is a fire, alarms are generated and transmitted to law enforcement, healthcare, and management departments. Moreover, E-FireNet achieves 0.98 accuracy, 1 precision, 0.99 recall, and 0.99 F1-score. A comprehensive investigation of various Convolutional Neural Network (CNN) models is conducted using the newly created Fire Surveillance SV-Fire dataset. The empirical results and comparison of numerous parameters establish that the proposed model shows convincing performance in terms of accuracy, model size, and execution time.

Keywords: Deep learning; drone; embedded vision; emergency monitoring; fire classification; fire detection; IoT; search and rescue

1 Introduction

In the last decade, drones have generated considerable attention as a remote sensing platform with a wide application range, including traffic control, disaster response, crop protection, and satellite image analysis [1,2]. Of late, incorporating a vision system, drone applications have been developed for monitoring, perceiving, and analyzing active and passive threats at the incident sites, for example, fire detection, flood threats, car accidents, and landslide-prone areas [3,4]. Additionally, drones can be rapidly deployed because their small size permits them to participate in mission-critical decisions, enabling better resource allocation and risk reduction. Drones are expected to function in disaster-affected areas, where connection to cloud services may not be effective and high-end equipment may not be readily accessible. To ensure operational performance and real analysis, a high-level of autonomy is required. Therefore, autonomous Unmanned Aerial Vehicles (UAVs), as well as Unmanned Ground Vehicles (UGVs), rely on their onboard sensors and embedded microchips for performing tasks rather than sending the data to a central control station. Furthermore, drones can cover a larger area within a shorter time span when combined with automatic route planning techniques, with onboard visual sensors as well as autonomous navigation. With the limited computational ability and power consumption, drones, however, present their own set of challenges [5].

CNNs and Deep Learning (DL), in particular, have been widely recognized as popular solutions for a broad range of applications based on computer vision, such as activity recognition, person recognition, vehicle recognition, and classification [6–10]. In prior research, utilizing transfer learning, a pre-trained CNN was used as a feature extricate, and certain layers have been added to perform classification for the current job, learning techniques to outperform standard machine learning methods using handcrafted features. Even though CNNs have proven to be more effective in classification, their inference time is high due to their high-computational power requirements when embedded in low-power devices, such as drones, which must perform multiple vision tasks simultaneously. For certain applications, a localized, integrated approach is more desirable over cloud processing due to security and privacy concerns [11]. In addition, tiny CNNs can ascertain the accuracy needed and the performance for specialized applications where substantial information does not exist and computational resource limits are enforced. Moreover, their training process is considerably easier, and they can be conveniently updated over air owing to their computational efficiency.

Soft computing methods based on Traditional Fire Warning Systems (TFWSs) [12] and optical sensors to prevent flames from spreading have been extensively researched and developed. Various scalar sensors, including ocular sensors, inferno sensors, as well as smoke sensors closer to the blazing fire [13] have been used in a TFWS for fire detection. However, scalar sensor-based solutions do not provide additional information regarding the area coverage, level of burning, location of the fire, or size of the fire. Additionally, the above-mentioned sensor systems require human intervention, such as a visit to the fire site in the event of a disaster. To overcome these limitations, various visual sensor-based methods have been proposed [14]. In surveillance systems, for the autonomous observation of fire catastrophes, traditional, DL, and vision-driven techniques play a key role in fire detection [15,16]. These algorithms offer a number of advantages such as rapid response, low-human-intervention requirements, cost-effectiveness, and greater coverage. However, traditional fire detection is difficult and time-consuming to process because it relies on hand-crafted feature extraction, and the procedures for constructing and evaluating the features are tedious. In particular, the monitoring of early fire and alarm generation, in traditional-based approaches is difficult because of the fluctuating lighting conditions, shadows, reflections, and low-detection accuracy. This study applies CNN models motivated by their potential in several areas, such as fire detection in surveillance footage [17]. However, DL includes an End-to-End (E2E) process for identifying features, which is computationally intensive and needs a considerable volume of training data. In this article, we have fine-tuned and proposed an efficient VGG-based (E-FireNet) model that has improved detection accuracy, has fewer parameters and can be deployed in actual scenarios. E-FireNet is not only capable of classifying fire and non-fire events but also looks for Object-of-Interest (OoI) located in the image. If the input image depicts a building fire or car fire, the model classifies it as a fire event on the specific object; if no fire is detected by the model, the resultant image is classified as non-fire. Furthermore, the baseline methods use computationally complex models that are not capable of being deployed on a drone, considering that a drone is a resource constraint device with limited processing power. Therefore, a lightweight CNN model that can be deployed in drones is highly desirable. The following are the significant contributions of this study:

- Early attempts for fire detection using pre-trained models with numerous parameters show limited performance when the variation in the data was low. To address these problems, this study presents a CNN-based framework for early fire detection in the images captured in diverse surveillance environments. The proposed E-FireNet model shows convincing performance with respect to the accuracy and time complexity for the Central Processing Unit (CPU) and Graphical Processing Unit (GPU), i.e., 0.98, 22.17, and 30.58 Frames Per Second (FPS), compared to renowned State-Of-The-Art (SOTA) models.
- As the existing fire datasets include limited scenarios and are monotonous, (i.e., considering only fire and non-fire scenarios), the generalization of the model is poor. Consequently, this study acquired a wide set of image samples containing real-world fire events such as building fire, car fire, and non-fire from various web sources, including social media platforms, NEWS sources, Google images, and YouTube videos.
- To verify the proposed method, this study conducted a comprehensive set of experiments over numerous pre-trained models such as NASNetMobile, MobileNetV1, EfficientNetB0, VGG19, and VGG16 using the newly generated Fire Surveillance (SV-Fire) dataset. Furthermore, to investigate the effectiveness, this article compared the performance of the proposed E-FireNet with the SOTA models with respect to the accuracy, parameters, and FPS.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature, Section 3 describes the proposed methodology, and Section 4 presents the experimental results. Finally, Section 5 summarizes the findings and suggests future directions.

2 Literature Review

Fire is an abnormal event that leads to serious injury and death, and affects precious resources within a short duration. Several techniques were proposed to monitor and control fire events in cities for saving life and property. However, fire detection in real-time is a challenging task. For instance, a CNN-based technique was proposed in [18], which aims to improve the accuracy and reduce the false-alarm rate. For improving the performance, a pre-trained model with fine-tuning of the uppermost layers was used. Furthermore, experiments were conducted over benchmark datasets, and 94.43% accuracy was achieved. Another technique was proposed [19], to realize a false-alarm system for a fire event. Through the transfer learning technique, InceptionV3 achieved excellent performance on test data. Several researchers trained models using satellite-captured images for the classification of fire and normal scenes. The main aim was to extract the region of fire using a local binary pattern for the reduction of false detection rates, and 98% accuracy was achieved.

The approach presented in [20] resolved the fire detection issue through classification and segmentation mechanisms. An artificial neural network was built for the binary classification and 76% accuracy was realized. Nevertheless, the segmentation method was applied to determine the fire border, whereas, for the fire mask, U-Net was used for up and down-sampling to obtain 92% precision and 84% recall. The technique introduced in [21] uses You-Look-Only-Once (YOLO) model for flame detection and extracts the visual features from video data frames. To overcome the overfitting problem and achieve efficient performance, augmentation techniques, such as rotation, flipping, and brightness adjustment were applied. Another study [22] used a Faster Region-based CNN (Fast-RCNN) to detect fire and normal scenes in an image, and later extracted the spatial features via a CNN; for temporal features, Long-Short Term Memory (LSTM) was employed to classify the target scene.

In a recent study [23], a fast and accurate algorithm was developed for extracting spatial features from surveillance video data. Three different versions of SqueezeNet were analyzed to compare their classification performances. In-depth experiments were conducted and 95.02%, 98.46%, and 98.52% accuracy were obtained with SqueezeNet1, SqueezeNet2, and SqueezeNet3, respectively. The technique presented in [24] applied different CNNs such as AlexNet, GoogLeNet, and VGG16 to recognize different events (smoke, non-fire, flame). The experimental results exhibited that the VGG16 model achieved the best performance. Similarly, in another approach [25], a lightweight CNN model was developed for flame detection in real-time, which mainly monitored and controlled the fire scenario in the early stage. A recent study [26] proposed a technique based on a deep saliency network for video-based smoke detection and compared the obtained results with those of ML and DL methods. In addition, the saliency network aimed to highlight the Region-of-Interest (RoI), i.e., the smoke area in the images. To further improve the model performance, various augmentation techniques were applied to the samples.

Several researchers with diverse background studies have applied lightweight models to detect and classify fire in real-time. For instance, [27] proposed a lightweight CNN for fire detection and classification. Furthermore, they computed the execution time to verify the model's adaptability in real-time processing.

Similarly, [28] introduced a method that could detect fire in real-time in both indoor and outdoor surveillance videos. A multi-expert system was first employed to collect data based on color, shape, and motion analysis. The Bag-of-Words (BoW) approach was then applied for motion representation. In addition, real-time and web scraping videos of fire were utilized in experiments where the proposed model achieved better performance. Researchers in [29] introduced a real-time fire detection technique using a fusion algorithm and several sensors (smoke, flame, and temperature) in indoor and outdoor domains for fire incident detection. The included literature is summarized and then listed in Table 1.

Table 1: Summary of the included literature, where BD and CD denote benchmark and custom dataset

Ref No.	Description	Dataset/ Type	Architecture	Scenario
[18]	Authors proposed architecture of deep learning for surveillance videos, inspired by GoogLeNet.	Foggia/BD	CNN	Outdoor and indoor
[19]	Authors proposed a novel method for forest fire classification based on satellite images.	NASA worldview/ CD	InceptionV3	Outdoor
[20]	Authors classified the presence and absence of fire in videos based on binary classification.	Fire flame/ CD	ANN	Outdoor
[21]	Authors applied the YOLO model for flame detection and compare the results of YOLO with the SOTA shallow learning method.	Fire flame/ CD	YOLO	Indoor
[22]	Authors used neural networks to detect fire and smoke in indoor and outdoor scenes in real-time using video data.	Fire and smoke/CD	RCNN, LSTM	Outdoor
[23]	Authors concatenated manual features with DL features to create fast and accurate smoke detection in a forest.	Smoke/CD	DL	Outdoor

(Continued)

Table 1 (continued)

Ref No.	Description	Dataset/ Type	Architecture	Scenario
[24]	Authors proposed a novel method to recognize video-based fire and smoke using the DL technique.	Fire and smoke/CD	CNN	Outdoor
[25]	Authors proposed a novel algorithm of CNN for real-time flame detection by pre-processing the fire videos.	Bilkent Uni. fire/BD	CNN	Indoor and outdoor
[26]	Authors proposed a method for real-time smoke detection in videos based on a deep saliency network.	Smoke/CD	Saliency Network	Indoor and outdoor
[27]	Authors presented a CNN architecture for fire detection in a surveillance scenario and compared the proposed method with SOTA techniques.	Foggia and BoWFire/ BD	CNN	Indoor and outdoor
[28]	Authors proposed a technique which is capable to detect fire in an early stage in real-time.	Fire flame/ CD	Multi-Expert System	Indoor and outdoor
[29]	Authors developed a fire detection robot based on sensors, such as smoke sensors, temperature semiconductor sensors, and ultraviolet sensors.	N/A	Multi-Sensor IoT System	Indoor

As part of this study, a CNN architecture E-FireNet is employed, for detecting fire. Several scenarios were examined in the custom SV-Fire dataset, such as a fire in a car, a fire in a building, and non-fire. In this study, the outdoor fire conditions differ from those described in previous literature. In comparison, prior studies targeted only wildfire scenarios, including fire, smoke, non-fire scenarios, and non-smoke scenarios.

3 Proposed Methodology

The proposed framework involves three main steps. In the first step, the collected fire images are pre-processed to increase the number of samples. In the second step, images of diverse classes are input to the proposed efficient CNN model, which effectively detects and classifies the fire into the respective class. In the final step, the model takes a decision based on the predicted label for the given input image. If the predicted label is a fire in a building or fire in a vehicle, an alert is generated to the nearest emergency response department to take early action. A detailed pictorial representation of the proposed framework is depicted in subsection 3.2; the step-wise procedure is presented in Algorithm 1. In the initialization step of Algorithm 1, the drone acquires a Video Stream (VS) and loads a pre-trained Fire Detection Model (FD_M). When the frame is read, the RoI_S is extracted and checked for the presence of fire. If it is a non-fire image, the next frame is selected; else, if a fire is detected in the frame, an alert is generated and sent to the emergency department and disaster teams. Furthermore, in the following sections, this article briefly discusses each step of the proposed framework.

Algorithm 1: Fire detection algorithm in a complex surveillance environment using E-FireNet.

```
Input: Drone Video Stream
Output: Return $\text{Fire}_{\text{Detected}}$
Initialization:
$\text{FD}_\text{M}$ ← Load Pre-Trained Fire Detection Model ($\text{FD}_\text{M}$)
VS ← Acquire Video Stream
while VS do
|       Frame ← Read (VS)
|       $\text{RoI}_\text{S}$ ← $\text{FD}_\text{M}$ (Frame)
|       if $\text{RoI}_{\text{Label}}$ is non-fire then
|       |       select next Frame;                /* No action processing next frame */
|       else
|       |       if $\text{RoI}_{\text{Label}}$ = fire then
|       |       |       Send an emergency alert;  /* Call disaster response team */
|       | end
|       end
end
Return: $\text{Fire}_{\text{Detected}}$
```

3.1 Pre-processing of the Collected Data

Pre-processing refers to all the alterations [30] performed on the raw data before being fed to the proposed E-FireNet model. Which is an E2E model that detects fire in a complex surveillance environment. To realize high-performance, DL models require immense training data; therefore, an augmentation technique is applied to generate new samples for training. For augmentation, several operations are performed, including different alignments, locales, and scales, as shown in Fig. 1. The applied augmentation techniques are the most efficient and convenient position augmentation in terms of upscaling the data samples. The experimental results before and after data augmentation are presented in subsection 4.3.

Furthermore, geometric transformations are applied to a normal image to obtain additional images from an input image. The input image was flipped horizontally and scaled. In addition, the input image was rotated clockwise and counter-clockwise by 10 and −10 deg. CNN architectures have become more resilient because of the usage of a large variety of samples, which improved the model classification capability. Thus, the model must be familiar with objects of various sizes, their alignment, and all types of data compositions. Therefore, a DL model must employ augmentation approaches for generating new images in order to deal with all these different attributes. During the augmentation process, the DL model learns the same object from different angles and viewpoints for better generalization.

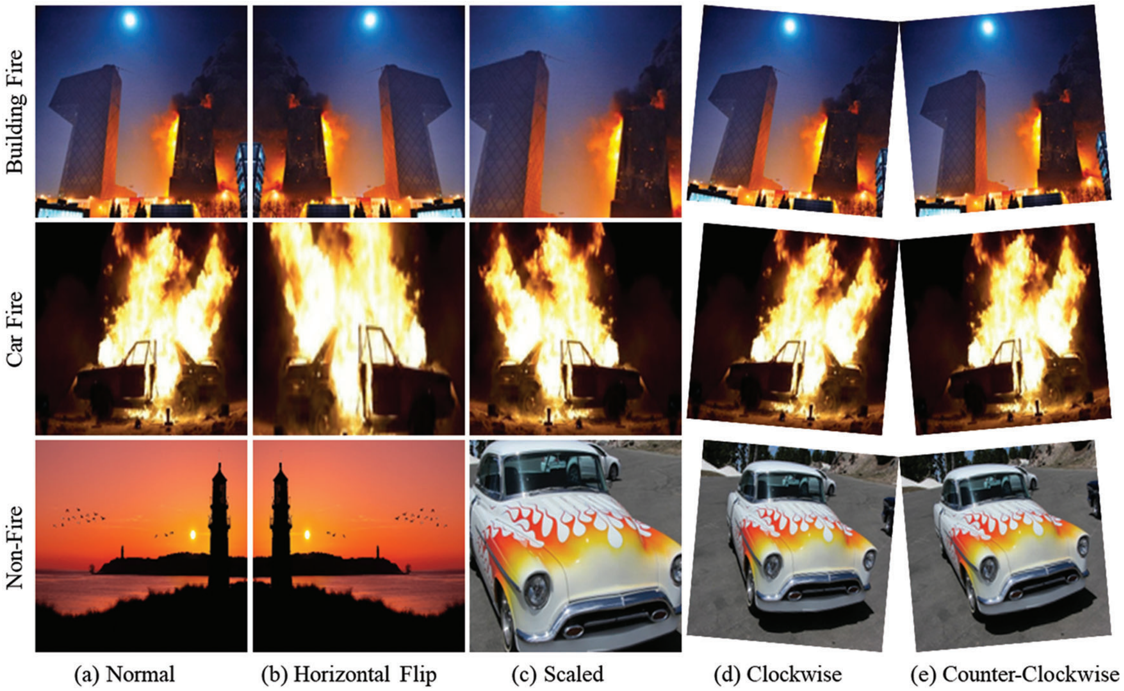


Figure 1: A variety of geometric transformations were undertaken in order to increase the number of samples in the dataset: (a) normal images, (b) horizontal flip, (c) scaling, and (d and e) rotation at various degrees

3.2 E-FireNet Framework

To monitor complex video surveillance, CNNs are often used for tasks such as activity and action recognition, anomaly detection, classification, and object detection [31–35], as well as a wide range of other identification, medical image diagnosis, video summarization, and segmentation applications [36–40]. The CNN architecture comprises three main components: the Convolution Layer (CL), pooling layer, and fully linked layer. A deep CNN includes a single input and a multitude of hidden, fully-linked, and Softmax layers. To build feature maps using deep CNNs, a number of parameters, local receptive fields, and various kernels are utilized that highlight the important characteristics of the objects in the picture. For dimensionality reduction, the feature maps are sub-sampled with average, minimum, or maximum pooling.

The selection of an appropriate CNN architecture for a certain situation is challenging in order to achieve adequate results while balancing the computational complexity. Each CNN has its own set of advantages and disadvantages based on the proposed architecture; for example, the design and development of AlexNet and VGG16 architectures are easier. In the ImageNet contest, AlexNet architecture was showcased and has become the benchmark architecture for DL. Increasing the number of CLs in a network is considered to enhance performance, as confirmed by the VGG model. As a robust feature extractor that can cope with large datasets and complex background identification tasks, the authors [24] suggested VGG16, a 16-layer architecture with the same filter size and considerable improvement in the classification.

Regardless of their numerous perks, VGG19 and VGG16 are not resource-friendly with respect to the overall size and training parameters. Architectures such as the NASNetMobile, MobileNetV1, and EfficientNetB0 CNN are resilient and considerably less costly, as MobileNetV1 and NASNetMobile have been specifically developed for fast inference time. Considering real-world implementation, resource computation cost, and repression of the constraints in present lightweight models, this study proposes an efficient fire detection and classification model, E-FireNet. The proposed framework is presented in Fig. 2. Initially, the performances of prominent ImageNet and pre-trained CNN architectures including VGG16, ResNet50, MobileNetV1, and NASNetMobile before developing the new E-FireNet framework are examined. This article particularly focuses on extracting fire zones using visually perceptible data successfully. As a result, this article included a smaller version of the captured image, unlike previous CNNs, to effectively recognize fire zones. This research entirely eliminated Block-5 of the VGG16 to reduce the number of parameters and training time. Additionally, the model was able to achieve higher accuracy when compared with other SOTA models despite a limited number of parameters and higher FPS. Moreover, this approach employs a smaller input size to retrieve minute information. As a result, the classifier can learn more characterized features.

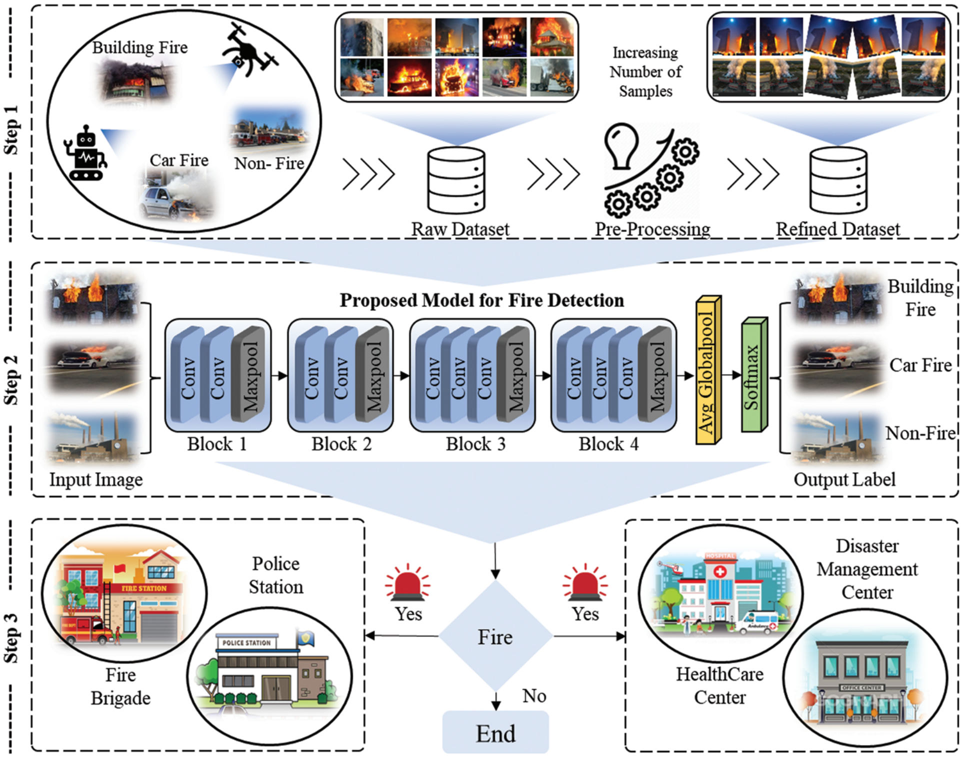


Figure 2: Proposed E-FireNet framework for fire detection and classification. Initially, pre-processing of the collected data is performed, followed by fire detection using the proposed technique, and finally, in case of fire detection, alarm generation to the law enforcement, healthcare, and management departments

The input image size for the proposed E-FireNet is 128 × 128 with 3 channels, and 32 distinct filters for red, green, and blue. Deep feature extraction can be accomplished, but the scale of each filter increases with respect to each progressive block. The filter sizes for the first, second, third, and fourth blocks were set as 64, 128, 256, and 512 respectively. In each layer of the proposed E-FireNet model, a linear function called Rectified Linear Activation (ReLU) is applied, which produces a direct output if the input is positive, otherwise, it produces zero. Subsequently, the input from the fourth block is forwarded to the pooling layers where Global Average Pooling is applied and it is finally conveyed to the Softmax layer, which provides a spread over the three class categories namely building fire, car fire, and non-fire. Table 5 In subsection 4.3 the article lists the training parameters of the proposed model.

4 Experimental Results

This section investigates the assessment measures and evaluation metrics in detail and describes the collected dataset and the graphical outcomes. The experimental setup and performance metrics are first described, followed by a discussion on the SV-Fire dataset, and the evaluation of the results. All the models, including the proposed E-FireNet, were trained using a total of 30 epochs with a low learning-rate to ensure that the model retained most of the previously learned knowledge. The pre-trained model progressively updates the learning parameters for optimum performance on the intended dataset. In subsection 4.3 the article compares the proposed model with the SOTA models and lists the main hyper-parameters utilized in these experiments. Based on the results, each model was retrained with its default input size, with a batch size of 16, and the Stochastic Gradient Descent (SGD) optimizer was equipped with a learning-rate and momentum of 1e-4 and 0.9, respectively. The experiments were conducted on an NVIDIA RTX 2060 Super GPU with 32-GB of onboard memory, a Keras DL framework, and TensorFlow for the back-end. As shown in the following equations, the performance of the proposed model was assessed by utilizing multiple evaluation metrics, including accuracy, precision, recall, and F1-score.

4.1 Evaluation Metrics

In the classification problem, accuracy is defined as the number of correct predictions produced by the model over all the types of predictions made,

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right), \tag{1}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

Precision is a metric that indicates the percentage of the dataset labeled as fire truly contains fire. The predicted positives (images predicted to be fire are TP and FP), and the photos with a fire scenario are TP.

$$Precision = \left(\frac{TP}{TP + FP}\right), \tag{2}$$

Recall is a metric that shows the percentage of observations in a dataset that were predicted as having a fire by the model. The real positives and fire images predicted by the model are TP.

$$Recall = \left(\frac{TP}{TP + FN}\right), \tag{3}$$

The F1-score measures the precision and recall harmonically.

$$F1 - score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right). \tag{4}$$

4.2 Dataset Collection

Finding appropriate data for evaluation is a difficult and time-consuming process. The authors could not find publicly accessible datasets for fire detection that satisfied the requirements for fire detection in buildings and cars. Owing to the unique nature of the findings of this study, a novel SV-Fire dataset is developed by collecting images from a variety of online sources. The major goal was to collect an image of a fire in a building and a car. Different settings and lighting situations are depicted in these high-resolution pictures. To make it more challenging, a new class of non-fire photos with an orange and red tint, as well as cars painted with fire decals, were added to the dataset. The overall statistics of SV-Fire are listed in Table 2.

Table 2: Overall statistics of the newly created SV-Fire dataset with a total of 1500 images: 1050 for training, 150 for testing, and 300 for validation

Dataset	Training	Testing	Validation	Total
Before augmentation	1050	150	300	1500
After augmentation	4200	600	1200	6000

This article presents several sample images as well as the general statistics of the newly created dataset. The code and dataset are publicly available at the following link: (https://github.com/NaqqashDilshad/E-FireNet). The total number of images in the SV-Fire dataset is 1500, while after augmentation the total reaches 6000. There are three subgroups in the SV-Fire dataset: training, validation, and testing. The training set comprises 70% of the total dataset, while the validation set comprises 20%, and the testing set is only 10%. A few instances from the recently collected dataset are presented in Fig. 3.

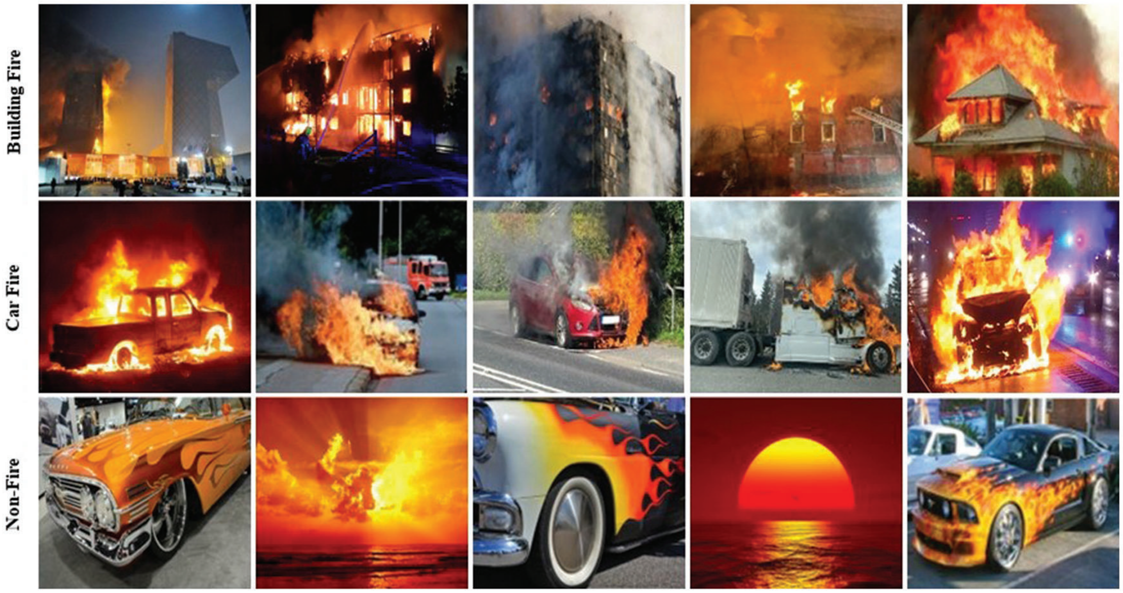


Figure 3: Sample images from our newly created SV-Fire dataset. The first, second, and third rows contain building fire, car fire, and non-fire images, respectively. Each row has five pictures. To make it more challenging, images with orange tint and fire look-alike are added

4.3 Performance Comparison with SOTA

This article compared the proposed model with different pre-trained CNN-based architectures for fire detection. The models were compared with respect to the number of parameters, precision, recall, F1-score, and accuracy as shown in Tables 3 and 4. NASNetMobile and MobileNetV1 have the least accuracy, whereas VGG16, VGG19, and the proposed E-FireNet model achieve high accuracies of 98%, 95%, and 98%, respectively. In addition, a comparison of the proposed model with MobileNetV1 shows that although both models are computationally efficient, the main difference is the accuracy where E-FireNet achieves approximately 21% higher accuracy than MobileNetV1.

Table 3: Overview of the comparison of the input size and network training parameters of the proposed E-FireNet with the SOTA models

Model	Input size	Batch size	Parameters (million)
NASNetMobile	224 × 224	16	4.27
MobileNetV1	224 × 224	16	3.22
EfficientNetB0	224 × 224	16	4.04
VGG19	224 × 224	16	139.58
VGG16	224 × 224	16	134.27
E-FireNet	128 × 128	16	7.63

Table 4: Evaluation of the proposed model E-FireNet against the SOTA models utilizing the SV-Fire dataset

Model	Class	Before augmentation				After augmentation			
		Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy
NASNetMobile	Car fire	0.55	0.54	0.55	0.59	0.64	0.44	0.52	0.59
	Building fire	0.54	0.73	0.62		0.58	0.58	0.58	
	Non-fire	0.69	0.51	0.59		0.56	0.74	0.64	
MobileNetV1	Car fire	0.59	0.54	0.57	0.62	0.75	0.77	0.76	0.77
	Building fire	0.53	0.58	0.55		0.75	0.70	0.73	
	Non-fire	0.72	0.72	0.72		0.81	0.84	0.83	
EfficientNetB0	Car fire	0.89	0.88	0.88	0.89	0.94	0.93	0.94	0.95
	Building fire	0.84	0.91	0.87		0.92	0.94	0.93	
	Non-fire	0.93	0.88	0.90		0.98	0.97	0.97	
VGG19	Car fire	0.92	0.92	0.92	0.92	0.99	0.88	0.93	0.95
	Building fire	0.91	0.89	0.9		0.90	0.99	0.94	
	Non-fire	0.93	0.95	0.94		0.98	0.99	0.99	
VGG16	Car fire	0.91	0.83	0.87	0.90	0.98	0.97	0.98	0.98
	Building fire	0.84	0.91	0.87		0.96	0.98	0.97	
	Non-fire	0.95	0.95	0.95		0.99	0.99	0.99	
E-FireNet	Car fire	0.82	0.77	0.80	0.81	0.98	0.96	0.97	0.98
	Building fire	0.71	0.78	074		0.95	0.99	0.97	
	Non-fire	0.89	0.88	0.88		1	0.99	0.99	

A comparison of the proposed model with VGG16 indicates that the results of VGG16 are proximate to those of the proposed model. However, the difference is the heavier weight, where VGG16 has 134.27 million parameters while E-FireNet has 7.63 million. The performance of the pre-trained models is listed in Table 4. It can be observed that the pre-trained models achieve high performance with a low false-alarm rate. However, the false prediction rate remains high and needs to be boosted. Therefore, this research explored a fine-tuned and pre-trained convolution neural network architecture (E-FireNet) with respect to accuracy and incorrect prediction. After tuning, E-FireNet attains the best performance among the other models with fewer false predictions.

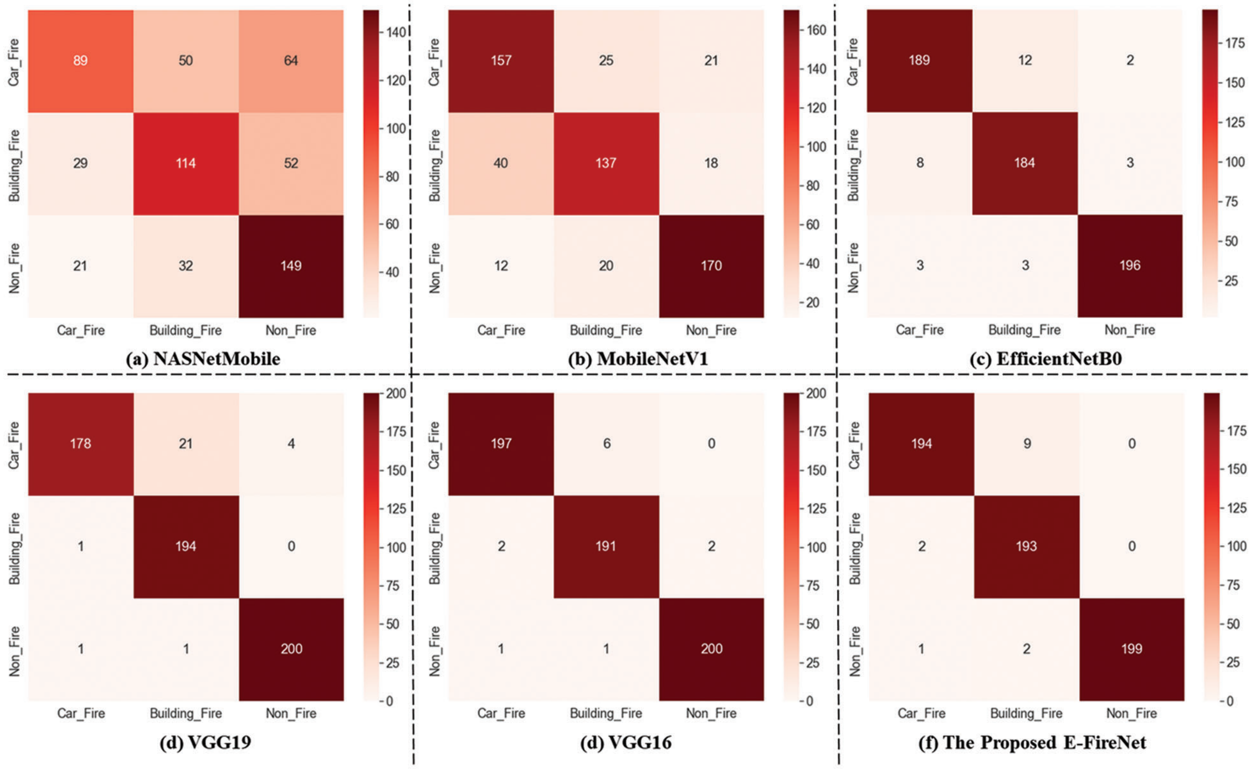
Table 5: The proposed E-FireNet summary with training parameters

Layer (type)	Filter	Kernel size	Stride	Parameters (million)
Conv_1	64	(3, 3)	1	0.001792
Conv_2	64	(3, 3)	1	0.036928
Max_Pool	–	(3, 3)	2	–
Conv_3	128	(3, 3)	1	0.073856
Conv_4	128	(3, 3)	1	0.147584
Max_Pool	–	(3, 3)	2	–
Conv_5	256	(3, 3)	1	0.295168
Conv_6	256	(3, 3)	1	0.59008
Conv_7	256	(3, 3)	1	0.59008
Max_Pool	–	(3, 3)	2	–
Conv_8	512	(3, 3)	1	1.18016
Conv_9	512	(3, 3)	1	2.359808
Conv_10	512	(3, 3)	1	2.359808
Max_Pool	–	(3, 3)	2	–
Global_Avg_Pool	512	–	–	–
Softmax (3)	–	–	–	0.001539
Total parameters	–	–	–	7.63

The confusion matrix for each SOTA model trained on the custom SV-Fire dataset is depicted in Fig. 4. The red diagonal correlates with TP, whereas the saturation represents the accurate classification. The proposed E-FireNet exhibits overall better classification accuracy compared to the SOTA models, although some of the images in all three categories (building fire, car fire, and non-fire) are misclassified. The training accuracy and training loss graphs are visualized in Fig. 5; the vertical axis represents the accuracy and loss, whereas the horizontal axis shows the total number of epochs. It is evident from Fig. 5 that E-FireNet is effective for fire detection. As the number of iterations of the training and validation processes increases, the training and validation accuracy line graph of the model change, as depicted in Fig. 5a. The proposed E-FireNet converges on 27 epochs, and the training and validation accuracies reach 100% and 98%, correspondingly. Likewise, the training and validation loss values change and drop to 0.0 and 0.09 respectively, as depicted in Fig. 5b.

4.4 Time Complexity Analysis

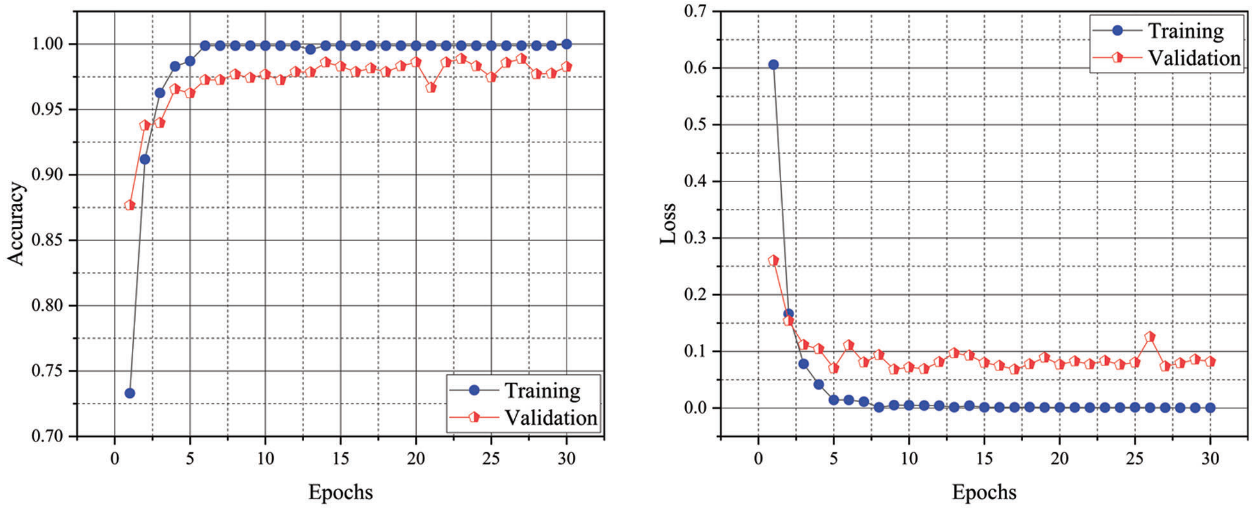
To assess a deep model's effectiveness, performance and deployment potential must be evaluated in real-time across various devices, including a CPU and GPU. The specifications of the CPU and GPU employed for analyzing the FPS of the proposed E-FireNet model are listed in Section 4. The criteria to assess the model performance for real-time application is that the model achieving 30 or more FPS is considered optimal for real-world scenarios. The FPS for the proposed E-FireNet model utilizing CPU and GPU is 22.17 and 30.58, respectively. Fig. 6 compares the proposed E-FireNet model in terms of the FPS with several baseline models.



(a) NASNetMobile (b) MobileNetV1 (c) EfficientNetB0

(d) VGG19 (d) VGG16 (f) The Proposed E-FireNet

Figure 4: Confusion matrices of the various CNN models against E-FireNet



(a) Model Accuracy (b) Model Loss

Figure 5: The proposed E-FireNet training accuracy and training loss. (a) Model accuracy (b) model loss

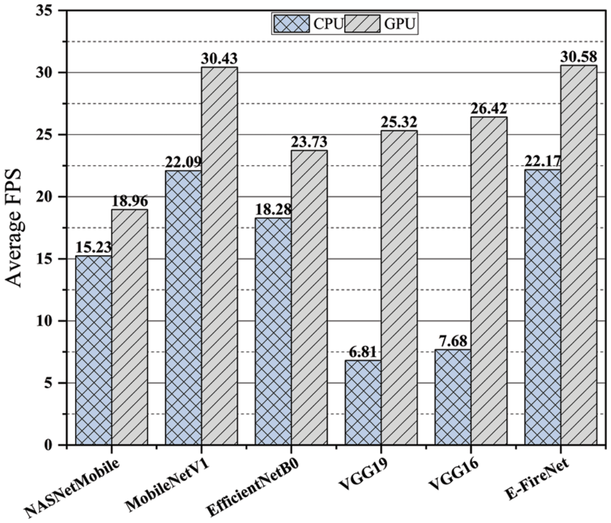


Figure 6: Comparison of the proposed E-FireNet with various deep models in terms of FPS

The experimental results show that employing the CPU and GPU, respectively, the FPS of the NASNetMobile model is 15.23, and 18.96, EfficientNetB0 model is 18.28 and 23.73, the VGG19 model is 6.81 and 25.32, VGG16 model is 7.68 and 26.42, and MobileNetV1 is 22.09 and 30.43. A comparison of the time complexity of the E-FireNet model with those of the other baseline models indicates that the performance of the proposed model is convincing. Thus, the proposed E-FireNet model is capable of real-time processing and operation.

5 Conclusion

To reduce social, environmental, and financial damage, CNN-based smart monitoring systems have been used to classify fire scenes in the early stages. Nevertheless, the research focuses on enhancing the accuracy, and attention to the model computation and generalization is limited. Therefore, this study presented an efficient framework (E-FireNet) that accurately classifies fire and non-fire images into their corresponding classes. The proposed E-FireNet achieves the best validation accuracy of 0.98 with limited parameters in comparison to the SOTA models. In addition, E-FireNet managed to achieve a precision of 1, a recall of 0.99, and an F1-score of 0.99. Furthermore, the SV-Fire dataset was collected since a dataset with diverse scenarios was not available for evaluating the proposed method. A set of experiments were performed using various CNN models and the proposed model, and their performances were compared in terms of accuracy, parameters, and FPS over two local systems (CPU and GPU) using the test data. Future research aims to expand the current dataset with new classes and apply vision transformers to fire detection.

Acknowledgement: This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No.2020-0-00959, Fast Intelligence Analysis HW/SW Engine Exploiting IoT Platform for Boosting On-device AI in 5G Environment).

Funding Statement: This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No.2020-0-00959).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Barmpoutis, P. Papaioannou, K. Dimitropoulos and N. Grammalidis, "A review on early forest fire detection systems using optical remote sensing," *Sensors*, vol. 20, no. 22, pp. 6442–6468, 2020.

[2] N. Dilshad, A. Ullah, J. Kim and J. Seo, "LocateUAV: Unmanned aerial vehicle location estimation via contextual analysis in an IoT environment," in *IEEE Internet of Things Journal*, pp. 1, 2022. DOI 10.1109/JIOT.2022.3162300.

[3] N. Dilshad, J. Hwang, J. Song and N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in *Proc. ICTC*, Jeju Island, SK, pp. 728–732, 2020.

[4] F. Khan, Z. Xu, J. Sun, F. Maula Khan, A. Ahmed *et al.,* "Recent advances in sensors for fire detection," *Sensors*, vol. 22, no. 9, pp. 3310–3334, 2022.

[5] H. Yar, T. Hussain, Z. Ahmad Khan, D. Koundal, M. Young Lee *et al.,* "Vision sensor-based real-time fire detection in resource-constrained IoT environments," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 21–36, 2021.

[6] I. Ullah Khan, S. Afzal and J. Weon Lee, "Human activity recognition via hybrid deep learning based model," *Sensors*, vol. 22, no. 1, pp. 323–339, 2022.

[7] S. Ullah Khan, I. Ul Haq, N. Khan, K. Muhammad, M. Hijji *et al.,* "Learning to rank: An intelligent system for person reidentification," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5924–5948, 2022.

[8] S. Ullah Khan, T. Hussain, A. Ullah and S. Wook Baik, "Deep-Reid: Deep features and autoencoder assisted image patching strategy for person re-identification in smart cities surveillance," *Multimedia Tools and Applications*, pp. 1–22, 2021.

[9] V. Kumar Yadav, A. Yadav, R. Yadav, A. Mittal, N. Hussain Wazir *et al.,* "A novel reconfiguration technique for improvement of pv reliability," *Renewable Energy*, vol. 182, pp. 508–520, 2022.

[10] R. Kumar, K. Bansal, A. Kumar, J. Yadav, M. Kumar Gupta *et al.,* "Renewable energy adoption: Design, development, and assessment of solar tree for the mountainous region," *International Journal of Energy Research*, vol. 46, no. 2, pp. 743–759, 2022.

[11] M. Al Mojamed, "Smart mina: LoraWAN technology for smart fire detection application for hajj pilgrimage," *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 259–272, 2022.

[12] S. Khan, K. Muhammad, S. Mumtaz, S. Wook Baik, V. H. C. de Albuquerque *et al.,* "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *Internet of Things Journal*, vol. 6, no. 6, pp. 9237–9245, 2019.

[13] A. S. Almasoud, "Intelligent deep learning enabled wild forest fire detection system," *Computer Systems Science and Engineering*, vol. 44, no. 2, pp. 1485–1498, 2022.

[14] Z. Yin, B. Wan, F. Yuan, X. Xia and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *Access*, vol. 5, pp. 18429–18438, 2017.

[15] H. A. Hosni Mahmoud, A. H. Alharbi and N. S. Alghamdi, "Time-efficient fire detection convolutional neural network coupled with transfer learning," *Intelligent Automation & Soft Computing*, vol. 31, no. 3, pp. 1393–1403, 2022.

[16] Q. An, X. Chen, J. Zhang, R. Shi, Y. Yang *et al.,* "A robust fire detection model via convolution neural networks for intelligent robot vision sensing," *Sensors*, vol. 22, no. 8, pp. 2929–2949, 2022.

[17] J. Sharma, O. Granmo, M. Olsen and J. Thomas Fidje, "Deep convolutional neural networks for fire detection in images," in *Proc. EANN*, Athens, GR, pp. 183–193, 2017.

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

[19] R. Shanmuga Priya and K. Vani, "Deep learning-based forest fire classification and detection in satellite images," in *Proc. ICoAC*, Chennai, IN, pp. 61–65, 2019.

[20] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé *et al.,* "Aerial imagery pile burn detection using deep learning: The flame dataset," *Computer Networks*, vol. 193, no. 4, pp. 108001–108011, 2021.

[21] D. Shen, X. Chen, M. Nguyen and W. Qi Yan, "Flame detection using deep learning," in *Proc. ICCAR*, Auckland, NZ, pp. 416–420, 2018.

[22] B. Kim and J. Lee, “A video-based fire detection using deep learning models,” *Applied Sciences*, vol. 9, no. 14, pp. 2862–2881, 2019.

[23] Y. Peng and Y. Wang, “Real-time forest smoke detection using hand-designed features and deep learning,” *Computers and Electronics in Agriculture*, vol. 167, pp. 105029–105047, 2019.

[24] G. Son, J. Park, B. Yoon and J. Song, “Video based smoke and flame detection using convolutional neural network,” in *SITIS*. Las Palmas, ESP, 365–368, 2018.

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

[26] G. Xu, Y. Zhang, Q. Zhang, G. Lin, Z. Wang *et al.,* “Video smoke detection based on deep saliency network,” *Fire Safety Journal*, vol. 105, pp. 277–285, 2019.

[27] K. Muhammad, S. Khan, M. Elhoseny, S. Hassan Ahmed and S. Wook Baik, “Efficient fire detection for uncertain surveillance environment,” *Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3113–3122, 2019.

[28] 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,” *Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1545–1556, 2015.

[29] R. C. Lou and K. Lan Su, “Autonomous fire-detection system using adaptive sensory fusion for intelligent security robot,” *Transactions on Mechatronics*, vol. 12, no. 3, pp. 274–281, 2007.

[30] A. Krizhevsky, I. Sutskever and G. Everest Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 60, no. 6, pp. 84–90, 2017.

[31] K. Muhammad, Mustaqeem, A. Ullah, A. Shariq Imran, M. Sajjad *et al.,* “Human action recognition using attention-based LSTM network with dilated CNN features,” *Future Generation Computer Systems*, vol. 125, no. 3, pp. 820–830, 2021.

[32] N. Khan, A. Ullah, I. U. Haq, V. G. Menon and S. W. Baik, “SD-Net: Understanding overcrowded scenes in real-time via an efficient dilated convolutional neural network,” *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1729–1743, 2021.

[33] N. Dilshad and J. Song, “Dual-stream siamese network for vehicle re-identification via dilated convolutional layers,” in *Proc. Smart IoT*, Jeju Island, SK, pp. 350–352, 2021.

[34] U. Ullah Khan, N. Dilshad, M. Husain Rehmani and T. Umer, “Fairness in cognitive radio networks: models, measurement methods, applications and future research directions,” *Journal of Network and Computer Applications*, vol. 73, pp. 12–26, 2016.

[35] A. Hussain, T. Hussain, W. Ullah and S. Wook Baik, “Vision transformer and deep sequence learning for human activity recognition in surveillance videos,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 22–32, 2022.

[36] A. Hussain, K. Muhammad, H. Ullah, A. Ullah, A. Shariq Imran *et al.,* “Anomaly based camera prioritization in large scale surveillance networks,” *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2171–2190, 2022.

[37] F. Mehmood, I. Ullah, S. Ahmad and D. Kim, “Object detection mechanism based on deep learning algorithm using embedded IoT devices for smart home appliances control in cot,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–17, 2019.

[38] S. Tiwari and A. Jain, “Convolutional capsule network for covid-19 detection using radiography images,” *International Journal of Imaging Systems and Technology*, vol. 31, no. 2, pp. 525–539, 2021.

[39] A. Jain, S. Tiwari, T. Choudhury and B. K. Dewangan, “Gradient and statistical features-based prediction system for covid-19 using chest x-ray images,” *International Journal of Computer Applications in Technology*, vol. 66, no. 3–4, pp. 362–373, 2021.

[40] S. Tiwari and A. Jain, “A lightweight capsule network architecture for detection of covid-19 from lung CT scans,” *International Journal of Imaging Systems and Technology*, vol. 32, no. 2, pp. 419–434, 2022.