Integrating Computer Vision: Advancements in Fire and Smoke Detection Using CNNs

Sanaz Janibeigi
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
sjanibeigi@gmail.com

Mahdi Bahaghighat
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
Bahaghighat@eng.ikiu.ac.ir
https://orcid.org/0000-0002-1813-8417

Zahra Yaghoubi
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
z.yaghoubi@eng.ikiu.ac.ir
https://orcid.org/0000-0001-7262-4611

Nastooh Taheri Javan
Computer Engineering Department, Imam Khomeini International University, Qazvin, Iran
nastooh@eng.ikiu.ac.ir
https://orcid.org/0000-0001-6928-3545

Abstract

Conventional fire detection systems, which depend on smoke and fire sensors, frequently have problems with calibration, susceptibility to environmental changes, and the requirement for ongoing maintenance. These flaws can result in unfavorable false alarms. Reducing the amount of damage caused by fire requires prompt and precise identification of potentially dangerous fire incidents. In response, this paper proposes a vision-based fire and smoke detection system based on computer vision and convolutional neural networks (CNNs). The main goal is to create a novel model that is built on three different architectures: Inception-v3, VGG-19, and VGG-16, each of which is customized by adjusting layers and hyperparameters. For a thorough assessment, the research integrates two publicly available datasets, BoWfire and Fire-Flame, and presents a brand-new NoFiSe dataset. Results show that the Inception-v3 model is quite effective; it outperforms other techniques with 91.11%, 94.99%, and 98.03% accuracy rates in the Fire-Flame, NoFiSe, and BoWfire datasets, respectively. The main novelty of this work is the integration of customized multi-scale CNN architectures with a newly developed multi-class fire/smoke dataset to achieve robust and generalized early fire detection under diverse real-world conditions. The study highlights the potential of intelligent fire suppression and emphasizes the need for continued research to improve current models in order to reduce future fatalities.

Keywords: Smoke detection, fire detection, early event systems, intelligent systems, CNN, Deep learning

1 Introduction

Human societies are considered to be at risk of fire, whether of human or natural origin. Fires tend to spread rapidly and can cause irreparable damage, including environmental degradation, deforestation, desertification, air pollution, economic loss, and even loss of life. The climate is one of the most important variables explaining fire outbreak patterns in arid regions. Dryland forest management has the potential to contribute to poverty alleviation and food security as long as it is assessed and managed sustainably [1]. There have been many attempts to detect fire over the years, and many industrial, commercial, and residential buildings have fire alarm systems. Generally, these systems work by using relevant sensors to detect smoke or heat, which requires proximity to the fire. There is a high error rate with such systems, and they may report a fire incorrectly as a result of environmental smoke that is not necessarily related to the fire itself (false alarm problem). Let us suppose that the sensors' sensitivity is reduced to reduce the number of false notifications. If this occurs, the system's accuracy will decrease and the problem of missed alarms will arise. Furthermore, in the real world, sensors fall out of the linear mode over time, which can exacerbate the issues already mentioned. Moreover, these systems can only reveal the event's occurrence, not its severity or exact physical location. To solve these problems, it is necessary to develop intelligent systems. Since closed-circuit cameras are increasingly being used in various places and digital technologies and image processing techniques are rapidly developing, there is a growing interest in replacing traditional fire and smoke detection methods with machine vision-based systems. In contrast to traditional systems, machine vision systems can be used both indoors and outdoors. As well as detecting its exact size and location, they can provide online information about the initial location of the fire. This includes smoke, the size of the fire, and its progress ([2, 3, 4]).

As a sub-field of Artificial Intelligence (AI), Machine Learning (ML) is defined as machines' ability to imitate intelligent human behavior. In ML systems, complicated tasks are carried out similarly to how people solve problems in everyday life ([5, 6]). In recent years, Deep Learning (DL) has become the most widely used field of machine learning. The favorable results demonstrate that DL can match or even surpass human performance. Convolutional Neural Networks (CNNs) are the most commonly used algorithm in deep learning, which have gained tremendous popularity due to their favourable performance ([7,8]). In surveillance systems, CNNs are widely utilized due to their appropriate application in image classification. By using intelligent monitoring systems, unusual events can be detected in a short period, such as fires, road accidents, and other incidents in which immediate action is required to prevent additional damage from occurring ([6, 8, 9]).

Although several CNN-based fire detection approaches have been proposed in recent years, many existing studies mainly focus on binary fire/non-fire classification under constrained environments and limited datasets. Furthermore, several methods rely on standard pretrained architectures without sufficient adaptation to the complex visual characteristics of fire and smoke, such as irregular shapes, dynamic textures, illumination variations, and background interference. In addition, the majority of available datasets do not simultaneously consider fire, smoke, and normal scenes within a unified framework, which limits the robustness and generalization capability of existing systems in real-world applications.

To address these limitations, this study proposes a customized computer vision-based fire and smoke detection framework utilizing optimized CNN architectures. The proposed approach integrates multi-class recognition of fire, smoke, and normal conditions while introducing a new challenging dataset, namely NoFiSe, containing diverse real-world environmental conditions. Moreover, unlike conventional implementations of pretrained CNN models, the proposed framework adapts and optimizes VGG-16, VGG-19, and Inception-v3 architectures to enhance feature extraction capability for complex fire and smoke patterns. Extensive experiments conducted on three benchmark datasets demonstrate the effectiveness and generalization ability of the proposed approach compared with existing methods. Here is a summary of the main contributions:

- Traditional sensor-based fire detection systems are highly sensitive to environmental conditions, which may lead to inaccurate detections and increased false alarm rates. Furthermore, these systems require strict and continuous maintenance to ensure reliable operation. To address these limítatenos, this study proposes an intelligent computer vision-based fire and smoke detection framework utilizing convolutional neural networks (CNNs) and transfer learning techniques to improve detection accuracy while enabling early and reliable event recognition.

- As early researchers worked on sensor-based models, their solutions were complex and expensive, especially in a broad range of applications. To cover large areas, these methods require a large number of sensors and must have different configurations for indoor and outdoor areas, while the proposed method based on computer vision is well-suited for deployment in a variety of environments.

- Our system integrates both fire detection and smoke detection problems into a standalone system that does not require any additional sensors for smoke detection. Not only can this approach reduce the overall project budget, but it can also reduce the rate of false alarms.

- CNN is integrated with some additional layers that can take into account both the local and contextual discriminating features of fire and smoke. These structures maintain sufficient levels of non-linearity that enables it to be adapted to different sources of fire and smoke.

- To verify the strength of our CNN-based model, we have gathered and created a brand-new dataset comprising 1663 images grouped into three categories: 'Fire', 'Smoke', and 'Normal'. Additionally, two public datasets were used to compare the detection scores for the multi-class classification problems.

The rest of the paper is organized as follows: In Section 2 of the paper, a brief description of the related work is provided. Section 3 provides technical information about the proposed model. Further, Section 4 presents a comprehensive empirical result based on three datasets. A summary of our work is presented in Section 5 along with a future research plan.

2 Related Works

Despite their advantages, sensor-based systems are not without disadvantages, including the need to be near the fire and higher false alarm rates, which can result in misdiagnosis and a loss of resources and time. These problems were addressed using visual systems. Due to the presence of surveillance systems and cameras in various environments, these systems have the advantage of being low-cost. In this way, it is possible to monitor a large area. In addition, it can eliminate the need to physically visit the site of a fire to confirm the fire's occurrence and the release of heat. Additionally, these systems are flexible enough to detect things other than fire, such as smoke and flames, and they can access information regarding the fire, such as its size, location, and intensity [2].

In this section, several research studies conducted in fire detection using visual systems have been reviewed. One of the earliest methods of detecting fire based on machine vision was color-based systems. In [10], color features were used to detect fire, rather than sensors. To differentiate fire-related pixels from the background, RGB and YCbCr color systems were used. A 90% accuracy rate has been reported for the proposed system.

A novel approach for fire detection in video streams was introduced by leveraging both motion characteristics and color information of fire [11]. Initially, moving objects were extracted using Gaussian Mixture Model (GMM)-based background subtraction. Subsequently, a multi-color detection strategy incorporating RGB, HSI, and YUV color spaces was applied to identify potential fire regions. The outputs of motion detection and color-based analysis were then integrated to accurately localize fire areas. Experimental evaluations on various fire video datasets demonstrated that the proposed method achieved improved effectiveness, robustness, and adaptability under different conditions.

[12] combined background subtraction for motion detection with CNN-based object detection and region classification to accurately identify fire regions while reducing computational complexity. To minimize false alarms, the authors further distinguished fire images from fire-like objects using a dedicated classification stage. Experiments conducted on a dataset of 5,075 fire-related images demonstrated a detection rate of 98.4%, a false

alarm rate of 99.9%, and an average processing time of 27.4 ms per frame, highlighting the suitability of the approach for real-time industrial fire monitoring applications. By increasing the number of hidden layers in an artificial neural network, more robust features could be learned from the input data. Several advances have been made in classification and recognition problems as a result of increased research into deep learning and neural networks ([7, 8, 9,13,14]).

Recent studies have demonstrated that video-based fire detection has become increasingly practical for fire disaster prevention due to advancements in deep convolutional neural networks (CNNs) and embedded processing hardware. However, many artificial intelligence (AI)-based approaches require substantial computational resources and high-performance graphical processing units (GPUs). To address this limitation, a cost-effective deep CNN architecture was proposed in [15] for real-time fire detection, optimized for the computational capabilities of the NVIDIA Jetson Nano platform. The study compared the proposed model with established CNN architectures, including AlexNet and SqueezeNet, and reported that the proposed network achieved a favorable balance between detection accuracy and computational efficiency while maintaining high accuracy and low loss values [15].

In [16], there are two different architectures used in this study, VGG-16, and ResNet-50. Aside from the regular models, the trained versions of these two architectures were also used due to their proper functioning. In this study, the best results were obtained with the trained architecture of ResNet-50, with an accuracy of 92.15%.

In [17], using a webcam, scientists developed a system that sends images to a laptop. A 42-layer Inception-v3 architecture was used in this study. There are fires and non-fires in the dataset, and the overall accuracy was 92%. When a fire is detected, a Telegram message is sent to the individual. Additionally, a smoke sensor has been employed to enhance accuracy.

To detect fire, scholars in [18] first extracted video frames and entered them into a database as training data and test data. A CNN was then used to label and train the training data. Based on YOLO architecture with nine convolution layers, 90% accuracy was reported in this study for predicting and detecting fires.

In [19], to replace traditional fire detection methods with methods based on deep learning, they designed a model based on CNNs. Since deep learning-based models require a large number of calculations, they stated that entering the images in their original size may increase learning time. Accordingly, in the proposed model, the images are first set to a fixed size and then provided as inputs to the model at a later stage. The purpose of this study is to design a novel CNN with nine layers, which is not a very sophisticated network. A final layer of the proposed CNN uses a classifier to classify the images into the fire and non-fire categories. Using the neural network proposed in this study, the accuracy was 92.5%.

Capsule neural networks combined with an adapted golden search optimizer improve pattern recognition on wildfire smoke imagery and BowFire, surpassing traditional feature-selection pipelines [20]. A lightweight CNN–attention backbone tailored to smoke plumes achieves higher average precision with fewer parameters and GFLOPs, emphasizing kernel design and attention for the early smoke detection system [21]. This study proposes deep learning-based forest fire detection using six pretrained models, achieving 99.74% and 98% accuracy on benchmark datasets, outperforming existing wildfire detection methods [22].

Object-detection frameworks have advanced parallel to classification. A hybrid YOLOv4–EfficientNetV2 + Efficient Channel Attention model achieves >93% mAP and outperforms YOLOv5–YOLOv9 on forest-fire imagery, showing the benefit of compound-scaled backbones with domain-aware preprocessing [23]. Comparative studies of YOLOv9–YOLOv11 on benchmark fire/smoke datasets report that small YOLOv11 variants yield the best trade-off between precision, recall, and mAP, especially for smoke in complex forest scenes [24]. [25] presents an AI-enabled IoT fire detection system integrating multi-sensor networks and machine learning for accurate real-time warnings, reduced false alarms, rapid emergency response, and automated prevention. Experimental results demonstrate high accuracy, low latency, scalability, and reliability for smart buildings and industrial safety applications mostly limited to indoor applications.

3 Methodology

This study aims to design an intelligent early event fire and smoke detection system based on computer vision and deep learning algorithms. As compared to previous studies in this field, many efforts have been made in our study to increase the accuracy of fire detection machines. To accomplish this, three architectures of trained Convolutional Neural Networks (CNNs) were utilized, with parameters being adjusted optimally. An overview of the presented models can be found in Figure 1.

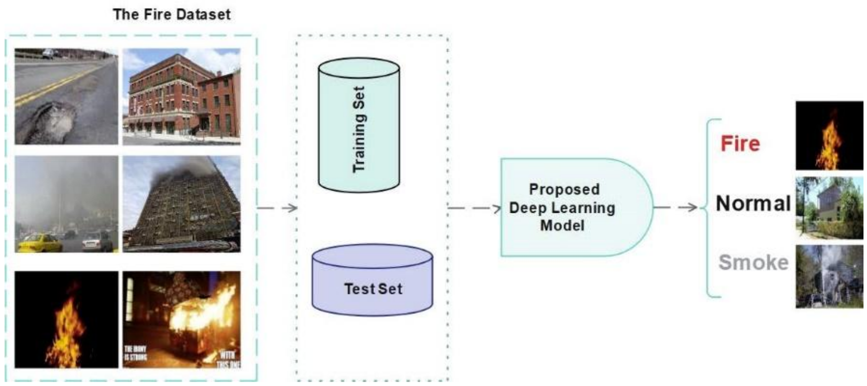


Figure 1: Our proposed model for fire & smoke detection

3.1 Dataset

This research utilized three datasets in general. We have gathered and created the first dataset, entitled the NoFiSe dataset, while the two other datasets are public sources. There are 1663 images in the NoFiSe collection, which is short for Normal & Fire & Smoke. In this collection, you will find images of fires that have occurred in accidents, buildings, urban spaces, roads, natural areas, and many other places. During the information gathering process, we have attempted to maintain a high degree of diversity in the dimensions, brightness, and resolution of images. In addition, we have considered diversity in the background and foreground of the images. According to Table 1, the first category contains 342 images with the 'Fire' label. The second category contains 921 images with the 'Normal' label, and the third category contains 400 images with the 'Smoke' label.

Table 1: Class distributions of our created dataset: NoFiSe.

	Class	No. of Samples
Our developed dataset: NoFiSe	Fire	342
	Normal	921
	Smoke	400

Figure 2 illustrates some of the images associated with each category.

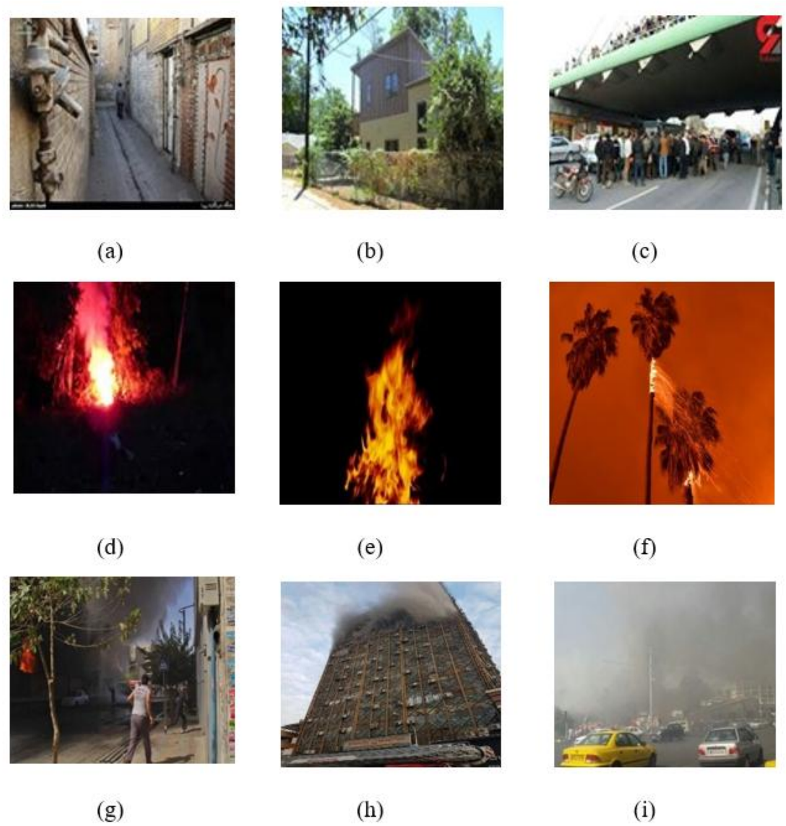


Figure 2: Three categories of the NoFiSe dataset; (a), (b), and (c): Normal, (d), (e), and (f): Fire, (g), (h), and (i): Smoke images.

In the second dataset, 2700 images were included in three categories. Table 2 provides information about this dataset.

Table 2: Class distributions in the Fire-Flame dataset [26]

	Class	No. of Samples
The Public dataset: Fire-Flame dataset	Fire	900
	Normal	900
	Smoke	900

Figure 3 illustrates examples of this dataset. We have used this dataset to evaluate the results of our research and it is a public one [26].

Figure 3: Three sample images from the Fire-Flame dataset [26]

The third dataset, called BoWfire, contains 226 images in two different classes: 'Fire' and 'Non-Fire'. Table 3 indicates that there are 119 images in the fire category and 107 in the non-fire category in this dataset. In addition to being a binary dataset, this dataset does not provide information on smoke detection.

Table 3: Class distributions in the BoWfire dataset [27]

	Class	No. of Samples
The public dataset: BoWfire dataset	Fire	119
	Normal	107

Despite this, the dataset can be useful in the field of fire detection since it contains challenging images such as red objects, sunlight, and others. A few examples of this dataset are shown in Figure 4 [27].

Figure 4: Two samples of the BoWfire dataset [27]

3.2 Convolutional Neural Networks (CNN)

An Artificial Neural Network (ANN) is designed to mimic the functional behavior of the human brain. ANNs are composed of a set of neurons connected in graphs. It is possible to define neurons as non-linear layer mathematical functions that are the basis for neural networks. There are some inputs and weights associated with neurons. Equation (1) multiplies the inputs of each neuron by its weights, and the result is added to the deviation. X represents the input, and w represents the weight [28].

$$T(x) = \phi(\sum_i W_i x_i + \mathrm{b}) = \phi\,(W.x + b) \qquad (1)$$

CNNs are similar to regular neural networks in that they are comprised of neural layers with learnable weights and biases. This neural network consists of several neurons. A variety of inputs are processed by neurons. The weight of the neurons is multiplied by the input; the result is then multiplied by bias and finally passed through an activation function. In contrast to other neural networks, CNNs receive different types of input, like images [28]. In a CNN, each layer is represented as a three-dimensional array of width, height, and depth. Figure

5 illustrates an overview of a CNN. There are four main layers in a CNN: the convolution layer, the nonlinear layer, the pooling layer, and the fully connected layer [29].

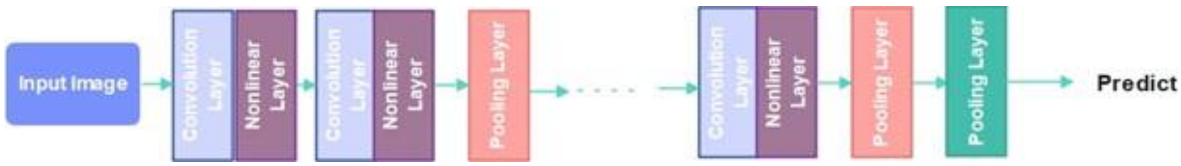


Figure 5: An overview of Convolutional Neural Networks

3.3 Convolution Layer

During the computation of a CNN, this layer is responsible for a significant portion of the computations. In the convolution layer, there are several filters. Convolution operations are performed between the filter and the layer's input, leading to the creation of a feature map. A CNN performs multiplication between the filter and input instead of multiplying the input and neuron weights internally. Filters are multidimensional arrays containing a set of numbers. A single pattern is stored primarily in filter numbers, and the filter is responsible for finding its pattern in the input layer ([29, 30]).

3.4 Non-linear layer

As with other artificial neural networks, CNNs also have activation functions. To increase flexibility, this function may be implemented as an independent layer in some cases. As a result of the activation function, the linear feature is destroyed in the network. In the natural world, most categories do not have clear linear boundaries. Therefore, it is necessary to simulate functions using a non-linearity layer. Sigmoid functions are used as activation functions and can take values between 0 and 1 (See Equation (2)). Training a network with a Sigmoid quickly reaches its saturation point, which indicates there is a problem with the Sigmoid function. Due to this, either the network learns very slowly or it does not learn at all. The Hyperbolic Tangent (Tanh) function can be considered an activator function. According to Equation (3), the output of this function falls within the range of 1 to 1, which is similar to the Sigmoid function expressed in coordinates. It is more common to use this function in binary classification. A popular activator function in neural networks is the unidirectional linear function. ReLU's calculation using Equation (4) is much faster than functions such as Sigmoid, and Tanh making its usage more common and less expensive for training deep networks ([30 ,31, 32]).

$$f(x)^{Sigmoid} = \frac{1}{1+e^{-x}} \qquad (2)$$

$$f(x)^{Tanh} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad (3)$$

$$f(x)^{ReLU} = \max(0, x) \qquad (4)$$

3.5 Pooling layer

CNN includes several layers, including the pooling layer. Using the pooling layer reduces the size of the feature map obtained by using the convolution layer. With the pooling layer, there are no trainable parameters, and maintaining the image's main features reduces the complexity of the network calculation and prevents overfitting. The merging process is similar to convolution in that a window moves over the image and simple sampling is performed. Maximum, average, and random pooling are the most common examples ([33, 34]).

3.6 Fully-connected layer

A fully connected layer in a CNN is generally located at the end of the structure and is used as a bundle. This layer performs the same function as a traditional multi-layer perceptron network. Multiple fully connected layers are usually used to increase the power of calculations. This layer contains some parameters as a result of the complete interconnection between the layers [30]. Each of these layers may have a different activation function. Typically, the last layer has a Softmax activation function. It is similar to the Sigmoid function in that it produces probabilities rather than values. It is widely used in classification applications to determine which category has the highest probability of matching the input. Considering the widespread use of convolution networks in classification applications, this function is included in the final layer of the network [35]. In addition, CNNs can be divided into two subcategories, namely feature extraction, and classification. Several convolutional and pooling layers are employed in the feature extraction sub-section, while several convolutional and pooling layers are employed in the classification sub-section, which uses a multi-layer perceptron. It consists of only fully connected layers [36].

3.7 Learning in CNNs

To train a CNN, one needs to set and update the values of filters in convolution layers as well as the weights in fully connected layers in such a way that the difference between the output prediction and the training data is as small as possible. As with multi-layer perceptron neural networks, CNNs are learned through an error backpropagation algorithm. Training neural networks are usually conducted using an algorithm based on backpropagation, in which loss functions and gradient reduction optimization algorithms play a fundamental role. A sample of training data is given to the network to calculate the output of this algorithm; then, using an error function, the network error is calculated (compared to the actual output), and an attempt is made to minimize the error by updating the learnable parameters (weights and network filters) so that the gradient is reduced.

3.8 Cost Function

Through forward propagation and the use of correct base labels, the loss function, also known as the cost function, measures the consistency of the output prediction. In classification problems, cross-entropy is used as a loss function, and in regression problems, mean squared error (MSE) is used as a loss function [30]. In Equation (5), cross-entropy is shown as a measure of dissimilarity between two distributions. The value of this function increases when the network experiences a high level of error [37].

$$Loss = -\sum_{c=1}^{M} y_{o,c} \log\left(p_{0,c}\right) \quad (5)$$

Where M is the number of classes, log represents the natural log, y is the binary indicator (0 or 1) if class label c is the correct classification for observation o, and finally, p denotes predicted probability observation o is of class c. Mean Squared Error (MSE) loss is also used in Machine Learning for classification & regression tasks. In mathematics, the MSE is defined as the difference between the predicted value (y^i) and the actual value (y_i). Equation (6) illustrates the MSE function. The higher the value of this function, the greater the difference between the original and the predicted value [38].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (6)$$

3.9 Gradient Descent

Gradient descent can be used as an optimization algorithm. To minimize the loss function in the direction of the negative gradient, this algorithm updates the learnable parameters repeatedly at an arbitrary step known as the learning rate. During gradient descent, a local minimum is attempted in the loss function. This algorithm uses the derivative of the function to determine the minimum since it is common to calculate the slope of the function to determine the minimum. For any learnable parameter, gradient reduction can be defined as the partial derivative of the loss function. A learning rate is indicated by the symbol W, while the loss function is indicated by the symbol L. To prevent the derivative from moving irregularly during training, the learning rate has a value between 0 and 1. Gradient reduction optimization is repeated until a reasonable minimum point has been reached. There have been many advancements in gradient reduction algorithms, including stochastic gradient descent (see Equation

(7)) with momentum (SGD) (Equation (8)) Adam (Equation (10) and Equation (11)) and RMSprop (Equation (12) and Equation (13)) [30].

$$\theta = \theta - \eta . \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \qquad (7)$$

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \qquad (8)$$

Where θ is:

$$\theta = \theta - v_t \qquad (9)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \qquad (10)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \qquad (11)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \qquad (12)$$

$$E[g^{\,2}]_t = 0.9E[g^{\,2}]_{t-1} + 0.1 g_t^2 \qquad (13)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{[g^{\,2}]_{t+\epsilon}}} g_t \qquad (14)$$

3.10 The Proposed models

Several solid pre-trained models are used in the development of the proposed model to achieve the desired results. The models used in this study were trained on large datasets. Our current study focuses on the classification of fire, smoke, and typical images. A suitable level of accuracy for fire detection and classification was achieved by combining three different architectures, VGG-16, VGG-19, and Inception-v3, on the NoFiSe dataset, BoWfire dataset, and Fire-Flame dataset. Using the trained models, the part of the model related to feature extraction has been reused, and the part related to classification has been redesigned and trained. After comparing these three architectures, we have finally concluded.

3.10.1 VGG-16 as the basic model

With 16 deep convolution layers, this model can recognize 1000 categories based on images in the Imagenet dataset. 224×224 pixels are the default input image size for the model, with three channels for RGB images. In step one of the architecture, 3×3 filters are used and in step two, maximum pooling is used [39]. As a first step, it is necessary to determine which middle layer should be used for feature extraction. The output of the last layer before the flattening operation, referred to as the bottleneck layer, is often used as a basis for the flattening operation [40]. Figure 6 illustrates the proposed research model using VGG-16's trained architecture. In this model, several layers have been added after the last layer located in block 5. To level the layer, the first layer must be fully connected. After the fully connected layer, there is a dropout layer with a rate of 0.2, followed by a fully connected layer with a ReLU function. To classify the images into three categories, a fully connected layer, and the maximum soft function are used.

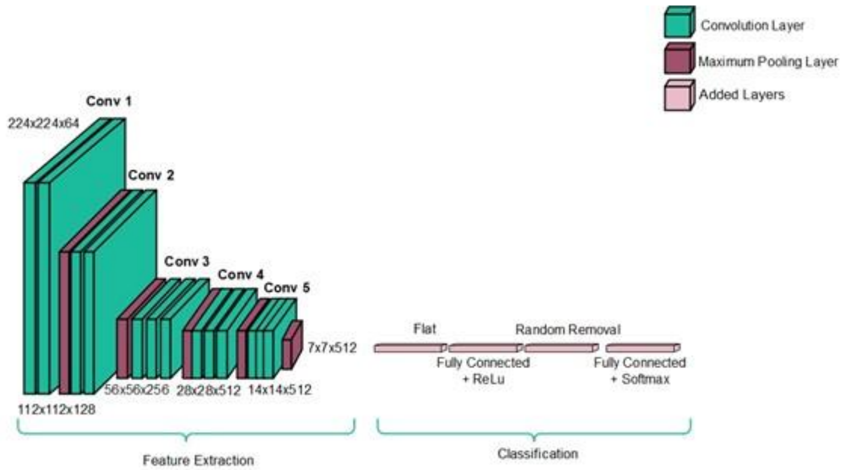


Figure 6: Proposed model found on VGG-16 architecture

3.10.2 VGG-19 as the basic model model

This network is presented with an image of dimensions 3×224×224 in the learned architecture. The VGG-19 architecture consists of 19 convolutional layers. As with the previous architecture, this architecture was trained using the dataset from the Imagenet challenge. The filters used were 3×3 with step 1 and maximum pooling with 2×2 window and step 2 [41]. Figure 7 illustrates the proposed model along with the added layers. There are four layers: flattening, fully connected layer with ReLU function, random removal with 0.2 rates, and finally, a fully connected layer with maximum soft function.

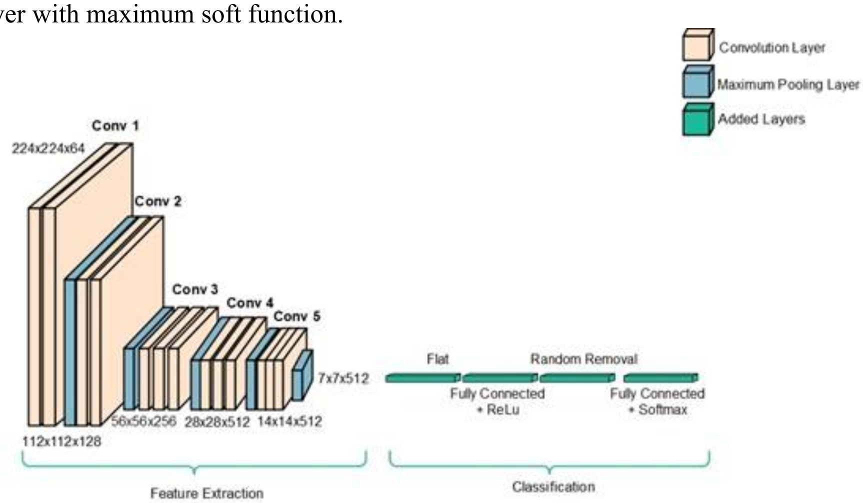


Figure 7: Proposed model found on VGG-19 architecture

3.10.3 Inception-v3 as the basic model

In 2015, this architecture was presented with 42 layers and a lower error percentage than previous versions. The overall architecture of this network consists of 11 Inception modules. One of the most significant changes in the third version is the factorization of the convolution layers. To reduce computational costs, smaller filter sizes, such as 3×3, have been used instead of larger filter sizes, such as 5×5 and 7×7. In this version, auxiliary classifiers are used to address the gradient fading problem [42]. The proposed research model is illustrated in Figure 8. This layer is designed to extract features from the Inception neural network version 3. There is a mixed layer of seven. The bottleneck layer is not the bottleneck of the network, but it can be used to maintain a sufficiently large (12×12) feature map, rather than a 5×5 feature map created with the bottleneck layer. The category that is fully connected is at the end of the list. Using this category, the output layer is first flattened to one dimension, followed by a fully connected layer with 1024 hidden units and a ReLU activation function. As a final activation function, the maximum-soft function is used, with the random removal rate set to 0.5.

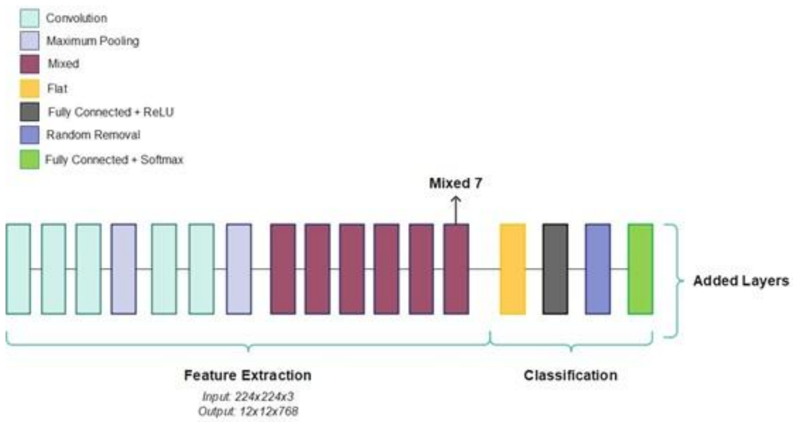


Figure 8: Proposed model with the Inception-v3 architecture

A model's accuracy is generally determined by how well it performs across all classes. When all classes are equally critical, this method is useful. It is calculated by dividing the number of correct predictions by the number of all predictions, as shown in Equation (15). This metric is used in this study to evaluate the results of the models.

$$Accuracy = \frac{True_{positive}+True_{negative}}{True_{positive}+True_{negative}+False_{positive}+False_{negative}} \tag{15}$$

The proposed method he proposed methodology differs from conventional CNN-based fire detection approaches in several important aspects. First, rather than limiting the problem to binary fire detection, the proposed framework performs multi-class classification involving fire, smoke, and normal scenes simultaneously. This design increases the practical applicability of the system in real-world monitoring environments where smoke may appear before visible flames.

Second, this study does not rely solely on off-the-shelf pretrained CNN architectures. Instead, the selected deep learning models, namely VGG-16, VGG-19, and Inception-v3, were customized through architectural modification and hyperparameter optimization to improve discriminative feature extraction. In particular, the Inception-v3 architecture was selected due to its capability to process multi-scale visual features using parallel convolution kernels, which is especially beneficial for detecting fire and smoke regions with varying shapes, scales, and textures.

Third, to improve robustness and evaluation diversity, a new dataset named NoFiSe was developed and integrated with two publicly available benchmark datasets. The proposed dataset includes challenging visual conditions such as complex backgrounds, illumination changes, smoke dispersion variations, and fire-like objects, thereby enabling a more realistic evaluation of the proposed framework.

4 Experimental Results

According to Table 4, the proposed model was implemented using the Python programming language. To implement the models, two parts of the dataset were used for model training and evaluation. There is an 80% training dataset and a 20% evaluation dataset. Basic configurations of the model can be seen in Table 4.

Table 4: Hardware & Software environments deployed in this study

OS	Windows 10-64bit
CPU	Intel(R) Core (TM)i7-7700HQ 2.8GHz
GPU	NVIDIA Tesla P100
RAM	16 GB
Programming language	Python 3.9
Software libraries	TensorFlow, Keras, OpenCV, and Scikit-Learn

Table 5: The basic configuration of some parameters in this study

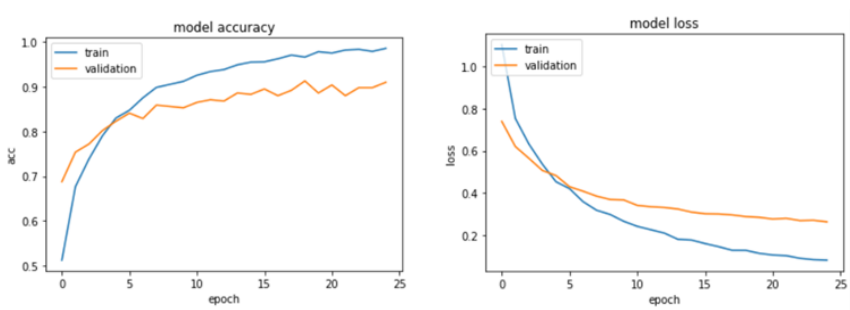
Parameters	Notes	Values
Sr	Data split ratio	0.8
Epochs	Epochs	25
BS	Batch size	150
LR	Learning rate	0.0001
Opt	Optimizer	RMSprop

4.1 Transfer learning and Fine-tuning

In this section, VGG-16, VGG-19, and Inception-v3 are implemented. Then, the results obtained from the architectures are compared with each other.

4.1.1 Result of VGG-16 based model

Using the modified VGG-16 architecture, the model trained on the NoFiSe dataset after 25 cycles achieved an accuracy of 90.99%. Figure 9 depicts the accuracy and loss functions of this architecture.

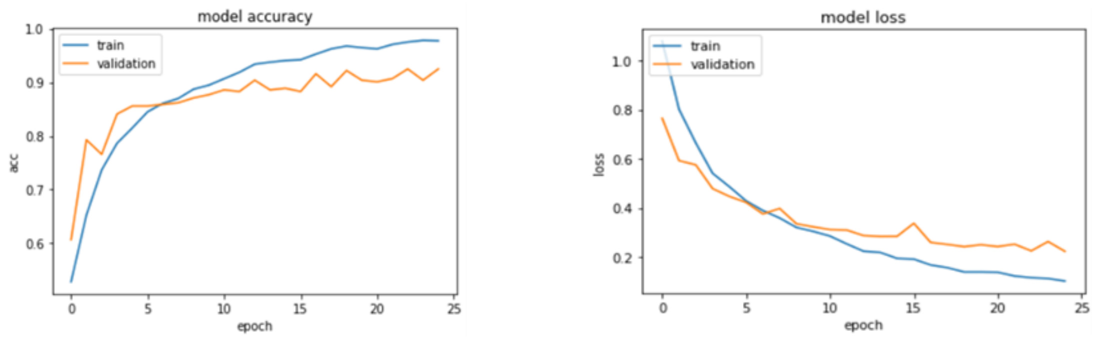


(a) (b)

Figure 9: Accuracy and loss functions of the designed model based on VGG-16 architecture. (a) Accuracy (b) Loss function

4.1.2 Result of VGG-19 based model

Using the NoFiSe dataset, this model was trained for 25 cycles and achieved an accuracy of 92.49%, which is slightly higher than the VGG- 16 architecture. Figure 10 illustrates the accuracy and loss function graphs of the model.

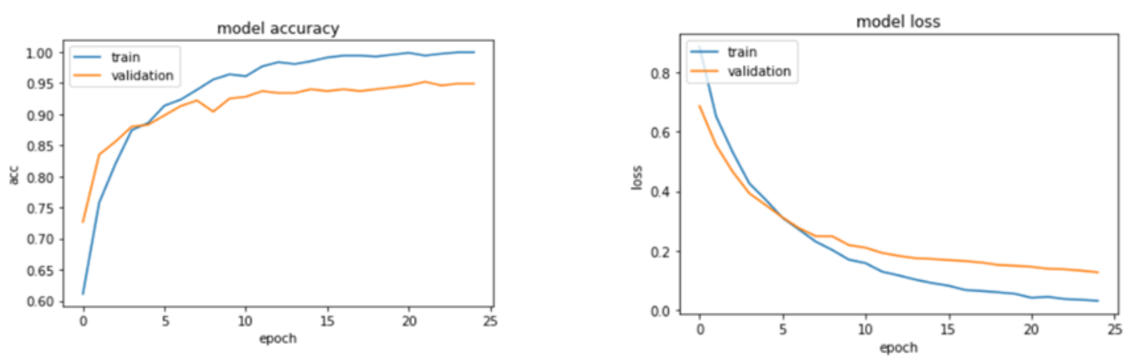


(a) (b)

Figure 10: Accuracy and loss functions of the designed model based on VGG-19 architecture. (a) Accuracy (b) Loss function

4.1.3 Result of Inception-v3 based model

The accuracy obtained for the NoFiSe dataset after 25 training sessions using this architecture, which is deeper than previous architectures (42 layers), is 94.99%. Consequently, the Inception model provides a higher level of accuracy than previous architectures. A graph of model accuracy is shown in Figure 11 (a) over a period of 25 training sessions. As shown in Figure 11 (b), the loss function for the model is also provided.



(a) (b)

Figure 11: Accuracy and loss functions of the designed model based on Inception-v3 architecture. (a) Accuracy (b) Loss function

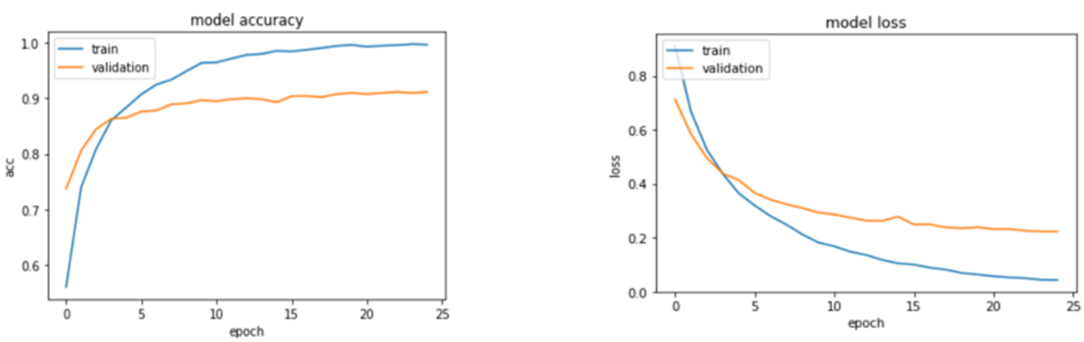
The accuracy of all three models is summarized in Table 6 Based on its enhanced accuracy than the model designed by the Inception architecture, this model is considered to be the basic model of the research. To ensure the proper performance of the model, we have included below the second and third datasets that have been used in fire detection studies using CNNs. Afterward, we compare the accuracy of our model with that of research models.

Table 6: Comparison of the accuracy obtained by the proposed architectures on the NoFiSe dataset

	Enhanced architecture	Accuracy (%)
NoFiSe dataset	VGG-16	90.99
	VGG-19	92.49
	Inception-v3	94.99

4.1.4 Evaluation of the Inception-v3 model with the Fire-Flame dataset

In this section, the proposed Inception-v3 model has been trained again using the Fire-Flame dataset, and its accuracy and loss functions are shown in Figure 12. This dataset contains 2700 images in three categories: fire, smoke, and normal images, used in [15]. As part of Gotthans's research, this dataset was used on three different architectures of CNNs, including SqueezeNet, AlexNet, and a proposed architecture called fire detection.



(a) (b)

Figure 12: Accuracy and loss functions of the proposed model on the Fire-Flame dataset. (a) Accuracy (b) Loss function

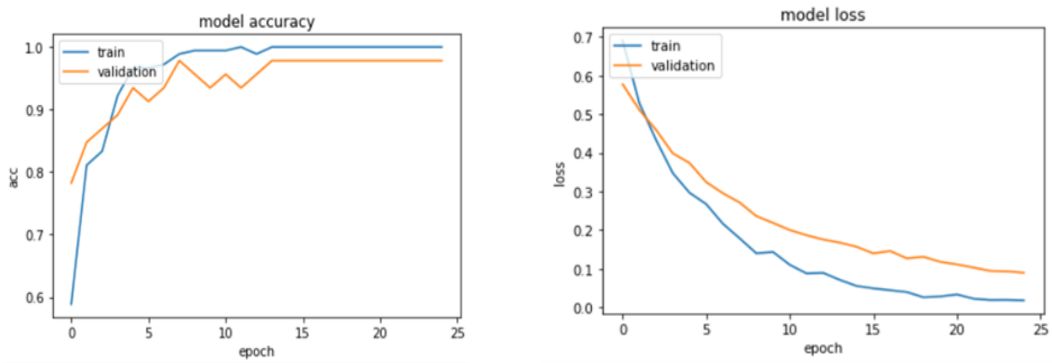
As shown in Table 7, the accuracy of these three architectures was compared with the accuracy obtained with the proposed research model for fire detection. The results obtained by the proposed model have shown an accuracy of 91.11%, which is much higher than the best accuracy of Gotthans's research about 81.66% in [15].

Table 7: Comparison of the accuracy achieved by proposed architectures based on Inception-v3 for the Fire-Flame dataset

Model	The basic architecture	Accuracy (%)
[15]	AlexNet	81.66
[15]	SqueezeNet	77.33
Proposed model	Batch size	91.11

4.1.5 Evaluation of the Inception-v3 model with the BoWfire dataset

BoWfire collection consists of only two categories, fire, and non-fire. This dataset has been used in many types of research [43, 44, 45, 46, 47]. Figure 13 presents the accuracy and loss diagrams of the proposed Inception-v3 model for the BoWfire dataset.



(a) (b)

Figure 13: Accuracy and loss functions of the proposed model on the BoWfire dataset. (a) Accuracy (b) Loss function

Besides, the results of several studies are presented in Table 8. It distinguishes this study with an accuracy of 98.03% from others in the literature.

Table 8: Comparing the accuracy of the proposed model with other studies for the BoWfire dataset

Model	Accuracy (%)
[43]	91.15
[44]	88.05
[45]	89.82
[46]	84.96
[47]	92.04
Proposed model	98.03

In Table 9, a general comparison has been made between the three datasets used in this study. From this comparison, it can be concluded that the proposed model has been able to provide a suitable improvement in the field of fire detection due to its better performance than previous studies.

Table 9: Summarizing assessment results: comparing the achievements of our model with previous studies based on three different datasets (NoFiSe, Fire-Flame, and BoWfire)

Dataset	Model	Accuracy (%)
NoFiSe	Proposed model	94.99
Fire-Flame [26]	[15]	81.66
Fire-Flame [26]	Proposed model	91.11
BoWfire [27]	[47]	92.04
BoWfire [27]	Proposed model	98.03

4.2 Discussion

The experimental results demonstrate that the proposed CNN-based framework achieved strong and consistent performance across all evaluated datasets. Among the investigated architectures, the customized Inception-v3 model produced the highest classification accuracies of 94.99%, 91.11%, and 98.03% on the NoFiSe, Fire-Flame, and BoWfire datasets, respectively. Besides, to improve the robustness of the proposed framework under realistic operating conditions, the NoFiSe dataset was designed to include diverse environmental scenarios involving variations in brightness, background complexity, smoke diffusion, object scale, and image quality. These considerations contribute to improving the generalization capability of the proposed system for practical intelligent surveillance applications. These findings indicate the effectiveness of multi-scale feature extraction for recognizing complex fire and smoke patterns under diverse environmental conditions. Furthermore, the results confirm that integrating fire, smoke, and normal-scene classification within a unified framework improves the practical applicability of intelligent surveillance systems. The superior performance achieved on both public and self-created datasets also demonstrates the robustness and generalization capability of the proposed approach compared with several existing CNN-based fire detection methods.

Although the proposed architectures achieved promising detection performance, practical deployment considerations were also taken into account during model development. The selection of VGG-16, VGG-19, and Inception-v3 enabled comparative analysis between model complexity and detection capability. In particular, Inception-v3 provides efficient multi-scale feature extraction while maintaining relatively lower computational overhead compared with deeper conventional CNN architectures. Nevertheless, real-time deployment in embedded systems or edge-AI platforms may still require additional optimization techniques such as model pruning, parameter quantization, lightweight CNN architectures, or hardware-aware acceleration strategies to reduce inference latency and memory consumption.

5. Conclusion and future works

Traditional sensor-based fire detection systems are highly sensitive to environmental conditions, which may lead to false alarms and require continuous maintenance. To address these limitations, this study proposed a computer vision-based fire and smoke detection framework using CNNs and transfer learning for accurate early event detection in both indoor and outdoor environments without requiring extensive sensor infrastructure.

Three deep learning architectures, namely VGG-16, VGG-19, and Inception-v3, were evaluated using the Fire-Flame, NoFiSe, and BoWfire datasets. Among them, the customized Inception-v3 model achieved the best performance, with accuracies of 91.11%, 94.99%, and 98.03%, respectively. The superior performance of Inception-v3 can be attributed to its multi-scale feature extraction capability, which enables more effective recognition of complex and variable-sized fire and smoke patterns compared with conventional CNN architectures.

The obtained results demonstrate that the proposed framework outperformed several existing studies while also validating the effectiveness of the newly developed NoFiSe dataset as a challenging benchmark for real-world fire and smoke detection applications.

Despite the strong performance achieved by the proposed framework, several practical deployment challenges should be considered for real-world intelligent surveillance systems. Deep CNN architectures, particularly Inception-v3, may require considerable computational resources for real-time inference, especially when deployed on embedded devices or edge-based monitoring platforms. In addition, environmental factors such as illumination variation, weather conditions, fog, occlusions, motion blur, and camera quality can influence detection stability and increase the difficulty of reliable fire and smoke recognition. Therefore, future research should investigate computationally efficient deployment strategies, including lightweight CNN architectures, model compression, pruning, quantization, and edge-AI optimization techniques. Such improvements could help achieve a better balance between detection accuracy, inference speed, and hardware efficiency while maintaining robustness in challenging real-world environments. Besides, we have not yet discussed other challenges such as exact localization of the fire/smoke place, predicting fire/smoke widespread and impact, forest fire spread speed estimation, etc. This information would be a great asset for the assessment and quantification of fire risk [48, 49]. Then, fire safety management can benefit from these risk factors [50]. It is believed that more research and efforts are needed to improve existing models, including the model presented in this study, to have intelligent fire suppression in the earliest stages. This could result in a significant reduction in the loss of life in the future.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Eskandari S, Chuvieco E (2015), Fire danger assessment in Iran based on geospatial information, International Journal of Applied Earth Observation and Geoinformation. 42, 57-64. https://doi.org/10.1016/j.jag.2015.05.006.

[2] Muhammad K, Ahmad J, Baik SW (2018) Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing. 288, 30-42. https://doi.org/10.1016/j.neucom.2017.04.083.

[3] Frizzi S, Kaabi R, Bouchouicha M, Ginoux JM, Moreau E, Fnaiech F (2016) Convolutional neural network for video fire and smoke detection, In IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. 877-882. https://doi.org/10.1109/IECON.2016.779319.

[4] Saeed F, Paul A, Karthigaikumar P, Nayyar A (2020) Convolutional neural network based early fire detection, Multimedia Tools and Applications. 79, 9083-99. https://doi.org/10.1007/s11042-019-07785-w.

[5] Mousavi S, Bahaghighat M (2025) Phishing website detection: An in-depth investigation of feature selection and deep learning, Expert Systems. 42(3), e13824. https://doi.org/10.1111/exsy.13824.

[6] Bahaghighat M, Akbari L, Ghasemi M, Xin Q (2026) Citation count prediction based on Google Scholar profiles and Clarivate's journal citation reports, Information Research an international electronic journal. 31(1), 46-69. https://doi.org/10.47989/ir31141005.

[7] Mohammadiha M, Bahaghighat M, Özen F (2025) Financial pattern recognition using feature extraction and classification, In 2025 IEEE 34th International Conference on Microelectronics (MIEL). 1-4. https://doi.org/10.1109/MIEL66332.2025.11261164.

[8] Rad PJ, Bahaghighat M (2024) Hierarchical text classification for Web of Science scientific fields, Facta Universitatis, Series: Electronics and Energetics. 37(4), 703-732. https://doi.org/10.2298/FUEE2404703J.

[9] Saffari S, Bahaghighat M (2025) Informatics and measurement in healthcare: deep learning for diabetic patient readmission prediction, Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska. 15(4), 56-64. https://doi.org/10.35784/iapgos.7835.

[10] Bakri NS, Adnan R, Ruslan FA (2018) A methodology for fire detection using colour pixel classification, In 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA), 94-98. https://doi.org/10.1109/CSPA.2018.8368692.

[11] Han XF, Jin JS, Wang MJ, Jiang W, Gao L, Xiao LP (2017) Video fire detection based on Gaussian Mixture Model and multi-color features, Signal, Image and Video Processing. 11, 1419-25. https://doi.org/10.1007/s11760-017-1102-y.

[12] Wu H, Wu D, Zhao J (2019) an intelligent fire detection approach through cameras based on computer vision methods, Process Safety and Environmental Protection. 127, 245-56. https://doi.org/10.1016/j.psep.2019.05.016.

[13] Jain R, Upreti K, Sharma A, Sen NK, Mathur AK (2026) An empirical comparative study of machine learning, deep learning, and transformer models for text classification, CLEI Electronic Journal. 29(2), 3-1. https://doi.org/10.19153/cleiej.29.2.3.

[14] Singh P, Nagpal S (2026) Hybrid RF-LSTM: An advance machine learning technique for prediction of air pollution in Indian cities, CLEI Electronic Journal. 29(2), 1-1. https://doi.org/10.19153/cleiej.29.2.1

[15] Gotthans J, Gotthans T, Marsalek R (2020) Deep convolutional neural network for fire detection, In 2020 30th international conference radioelektronika (RADIOELEKTRONIKA), 1-6. https://doi.org/10.1109/RADIOELEKTRONIKA49387.2020.9092344.

[16] Sharma J, Granmo OC, Goodwin M, Fidje JT (2017) Deep convolutional neural networks for fire detection in images, In Engineering Applications of Neural Networks: 18th International Conference, EANN. 183-193. https://doi.org/10.1007/978-3-319-65172-9_16.

[17] Sadewa RP, Irawan B, Setianingsih C (2019) Fire detection using image processing techniques with convolutional neural networks, In 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). 290-295. https://doi.org/10.1109/ISRITI48646.2019.9034642.

[18] Lestari DP, Kosasih R, Handhika T, Sari I, Fahrurozi A (2019) Fire hotspots detection system on CCTV videos using you only look once (YOLO) method and tiny YOLO model for high buildings evacuation, In 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE). 87-92. https://doi.org/10.1109/IC2IE47452.2019.8940842.

[19] Hao X, Linhu C, Weixin H (2019) Video Flame Detection Using Convolutional Neural Networks, In 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC). 539-543. https://doi.org/10.1109/ICIVC47709.2019.8981110.

[20] Liu, L., Chen, L., & Asadi, M. (2025). Capsule neural network and adapted golden search optimizer based forest fire and smoke detection. Scientific Reports, 15. https://doi.org/10.1038/s41598-024-81742-y.

[21] Hoang, Q., Hoang, Q., & Oh, H. (2025). An Efficient Forest Smoke Detection Approach Using Convolutional Neural Networks and Attention Mechanisms. Journal of Imaging, 11. https://doi.org/10.3390/jimaging11020067.

[22] Khirani, S., Souahlia, A., Rabehi, A., Bourennane, M., Guermoui, M., Tibermacine, I. E., & Rabehi, A. (2026). Advanced evaluation of pre-trained CNN models for accurate forest fire detection. Natural Hazards, 122(6), 234. https://doi.org/10.1007/s11069-026-07976-3

[23] Mamadmurodov, A., Umirzakova, S., Rakhimov, M., Kutlimuratov, A., Temirov, Z., Nasimov, R., Meliboev, A., Abdusalomov, A., & Cho, Y. (2025). A Hybrid Deep Learning Model for Early Forest Fire Detection. Forests. https://doi.org/10.3390/f16050863.

[24] Alkhammash, E. (2025). A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection. Fire. https://doi.org/10.3390/fire8010026.

[25] Swetha, P. B., Vennakandla, C. S. R., Sonti, S. R., Periasamy, P. S., Naresh, A., & Gali, S. (2026, February). Real-Time Smart Fire Detection and Prevention System Using IoT Sensors and ML Algorithms. In 2026 International Conference on Visual Analytics and Data Visualization (ICVADV) (pp. 1330-1336). IEEE. https://doi.org/10.1109/ICVADV67766.2026.11470528

[26] DeepQuestAI, Fire-Smoke dataset. https://github.com/DeepQuestAI/Fire-Smoke-Dataset, 2022

[27] Bitbucket, Bowfire dataset. https://bitbucket.org/gbdi/bowfire-dataset/downloads/, 2022

[28] Zhou SK, Rueckert D, Fichtinger G, Handbook of medical image computing and computer assisted intervention, first ed., ISBN: 9780128161760, Academic Press, London, 2019.

[29] Pinaya WH, Vieira S, Garcia-Dias R, Mechelli A, Convolutional neural networks, in: Machine learning, Academic Press, 2020, pp. 173-19. https://doi.org/10.1016/B978-0-12-815739-8.00010-9

[30] Aggarwal, C. C. (2018). Neural networks and deep learning (Vol. 10, No. 978, p. 3). Cham: Springer. https://doi.org/10.1007/978-3-031-29642-0

[31] Szandała T, Review and comparison of commonly used activation functions for deep neural networks, Bio-inspired neurocomputing. (2021) 203-24. https://doi.org/10.1007/978-981-15-5495-7_11.

[32] Li Z, Liu F, Yang W, Peng S, Zhou J (2021) A survey of convolutional neural networks: analysis, applications, and prospects, IEEE transactions on neural networks and learning systems. https://doi.org/10.1109/TNNLS.2021.3084827.

[33] Tian Y (2020) Artificial intelligence image recognition method based on convolutional neural network algorithm, IEEE Access. 8, 125731-44. https://doi.org/10.1109/ACCESS.2020.3006097.

[34] Sze V, Chen YH, Yang TJ, Emer JS (2017) Efficient processing of deep neural networks: A tutorial and survey, Proceedings of the IEEE. 105(12), 2295-329. https://doi.org/10.1109/JPROC.2017.2761740.

[35] Demir F, Abdullah DA, Sengur A (2020) A new deep CNN model for environmental sound classification, IEEE Access. 8, 66529-37. https://doi.org/10.1109/ACCESS.2020.2984903.

[36] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C (2018) A survey on deep transfer learning, In Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks. 270-279. https://doi.org/10.1007/978-3-030-01424-7_27.

[37] Wang Q, Ma Y, Zhao K, Tian Y (2020) A comprehensive survey of loss functions in machine learning, Annals of Data Science. 1-26. https://doi.org/10.1007/s40745-020-00253-5.

[38] Nadipally M, Optimization of methods for image-texture segmentation using ant colony optimization, in: Intelligent data analysis for biomedical applications, Academic Press, 2019, pp. 21-47. https://doi.org/10.1016/B978-0-12-815553-0.00002-1

[39] Rangarajan AK, Purushothaman R, Ramesh A (2018) Tomato crop disease classification using pre-trained deep learning algorithm, Procedia computer science. 133, 1040-7. https://doi.org/10.1016/j.procs.2018.07.070.

[40] Harris E, Mihai D, Hare J (2021) How convolutional neural network architecture biases learned opponency and color tuning, Neural Computation. 33(4), 858-98. https://doi.org/10.1162/neco_a_01356.

[41] Wen L, Li X, Li X, Gao L (2019) A new transfer learning based on VGG-19 network for fault diagnosis, In 2019 IEEE 23rd international conference on computer supported cooperative work in design (CSCWD). 205-209. https://doi.org/10.1109/CSCWD.2019.8791884.

[42] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision, In Proceedings of the IEEE conference on computer vision and pattern recognition. 2818-2826. https://doi.org/10.1109/CVPR.2016.308.

[43] Jadon A, Varshney A, Ansari MS (2020) Low-complexity high-performance deep learning model for real-time low-cost embedded fire detection systems, Procedia Computer Science. 171, 418-26. https://doi.org/10.1016/j.procs.2020.04.044.

[44] Muhammad K, Ahmad J, Baik SW (2018) Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing. 288, 30-42. https://doi.org/10.1016/j.neucom.2017.04.083.

[45] Muhammad K, Ahmad J, Lv Z, Bellavista P, Yang P, Baik SW (2018) Efficient deep CNN-based fire detection and localization in video surveillance applications, IEEE Transactions on Systems, Man, and Cybernetics: Systems. 49(7), 419-34. https://doi.org/10.1109/TSMC.2018.2830099.

[46] Muhammad K, Ahmad J, Mehmood I, Rho S, Baik SW (2018) Convolutional neural networks based fire detection in surveillance videos, IEEE Access. 6, 18174-83. https://doi.org/10.1109/ACCESS.2018.2812835

[47] Muhammad K, Khan S, Elhoseny M, Ahmed SH, Baik SW (2019) Efficient fire detection for uncertain surveillance environment, IEEE Transactions on Industrial Informatics. 15(5), 3113-22. https://doi.org/10.1109/TII.2019.2897594.

[48] Gibson L, Adeleke A, Hadden R, Rush D (2021) Spatial metrics from LiDAR roof mapping for fire spread risk assessment of informal settlements in Cape Town, South Africa, Fire Safety Journal. 120, 103053. https://doi.org/10.1016/j.firesaf.2020.103053.

[49] Lau CK, Lai KK, Lee YP, Du J (2015) Fire risk assessment with scoring system, using the support vector machine approach, Fire Safety Journal. 78, 188-95. https://doi.org/10.1016/j.firesaf.2015.10.003.

[50] Lim DH, Na WJ, Hong WH, Bae YH (2023) Development of a fire prediction model at the urban planning stage: Ordinary least squares regression analysis of the area of urban land use and fire damage data in South Korea, Fire Safety Journal. 103761. https://doi.org/10.1016/j.firesaf.2023.103761