LoCoNet: A Low-Complexity Convolutional Neural Network Model for Efficient Fire Detection in Outdoor Environments

Arwa M. Taqi
Department of Electrical and Electronic Engineering, College of Engineering, University of Kerbala, Iraq
arwa.h@uokerbala.edu.iq

Hameed R. Farhan
Department of Electrical and Electronic Engineering, College of Engineering, University of Kerbala, Iraq
hameed.r.f@uokerbala.edu.iq (corresponding author)

Ahmed Y. Awad
Department of Communication and Mobile Computing, College of Engineering, Alzahraa University for Women, Kerbala, Iraq
ahmed.yahya.awad@alzahraa.edu.iq

Received: 23 January 2025 | Revised: 22 February 2025 | Accepted: 28 February 2025

 DOI: https://doi.org/10.48084/etasr.10322

ABSTRACT

Early Fire Detection (FD) is essential, yet preventing damage to human life and property presents challenges. This study introduces a reliable and fast FD framework using a new Convolutional Neural Network (CNN) model called Low-Complexity Network (LoCoNet). The LoCoNet model deals with color images of 24×24 pixels, highly decreasing memory usage and processing time. The structure of the LoCoNet model consists of three convolutional layers, each utilizing a kernel size of 1×1, followed by a max-pooling layer, effectively halving the data size. Next, a flattening layer transforms the data into a 1-D vector. Then, a fully connected dense layer follows, and a dropout layer randomly deactivates 50% of its neurons during training. Finally, the output layer classifies the images according to the probability of fires occurring, predicting whether there are fires. K-fold cross-validation with various *K* values divided the dataset into training and testing sets. Multiple CNN models were investigated, and their results were compared to estimate their performance. According to the experimental results, the proposed LoCoNet model surpasses others in accuracy, processing speed, and memory usage, achieving an accuracy of approximately 99%, consuming about 2.86 s in model training, and using only 81.25 KB of memory. Compared to related approaches, the proposed LoCoNet model significantly decreases computational complexity while achieving high accuracy with minimal processing time.

Keywords-Convolutional Neural Network (CNN); cross-validation; deep learning; fire detection; low-complexity system; outdoor environment

I. INTRODUCTION

Fires cause numerous deaths, injuries, and losses in money and possessions worldwide every year. If not detected and treated early, fire hazards can significantly increase their criticality to human life or personal and public property. Fire Detection (FD) can be achieved using sensors or image processing techniques [1]. FD systems are invented to identify earlier scenes of a fire's growth, allowing for the safe evacuation of occupants. Early detection protects emergency responders and public property [2, 3]. When a small fire can overtake a building, forest, or vehicle in minutes, it is essential to have multiple fire detectors placed to help detect a fire quickly [4]. Fire detectors are necessary for homes, shopping malls, hotels, vehicles, factories, gardens, farms, forests, and streets. Their importance becomes clear when humans are out of place, sleeping, or cannot observe the entire area [5]. Sensors are utilized to detect fires and make decisions accordingly. However, many existing sensors, including smoke, flame, and heat detectors, tend to respond slowly [6]. They must also be strategically placed in different places to cover all essential areas. However, they cause some issues that make them unsuitable for outdoor use, as they cannot identify the fire's site and sometimes trigger false alarms [7, 8]. Accurate FD has gained considerable attention due to the challenges associated with traditional methods. However, refining FD technologies, including improved sensors, detection tools, and video-based FD, offers a promising future for fire safety [9]. Computer vision-based systems have recently replaced conventional FD methods due to rapid advances in video processing and digital camera techniques. Furthermore, remote sensing technologies have been used in recent decades to monitor open areas and land, and satellite imagery systems are sometimes used. These systems use color cues, flame pixel motion, and edge detection to identify flames [10].

A method that relies on colors to detect fires may fail when images contain objects similar to flame and smoke colors, such as sunlight, light sources, fog, and dust. Therefore, the use of different strategies may reduce inaccurate detections. Deep Learning (DL) has succeeded in different fields, especially FD, where various CNN models have recently been employed. The drawbacks of these models arise from their complex structures, especially the large number of layers and parameters. This increased complexity leads to higher memory usage and slower processing speeds, which can prolong processing times to several hours, as observed in the DenseNet201 model. The proposed work contributes to reducing system complexity and maintaining high accuracy. Unlike tasks that require recognizing detailed patterns, such as identifying faces [11], where a larger image size is essential, detecting fires with supervised DL does not require such a large size, and a smaller size is sufficient. As a result, the proposed LoCoNet model accepts images that are only 24×24 pixels in size, which is relatively uncommon in other models.

Researchers have proposed various methods for detecting fires using image processing, indoors or outdoors. Due to the topic's significance and complexity, it is still being examined. Due to the numerous FD approaches, it is difficult to mention all of them, so some recent and diverse proposed schemes are highlighted. In [12], different datasets for indoor and outdoor fires were used to detect smoke and flames. Four types of CNN were compared for object detection, concluding that YOLOv3 was the best, achieving 83.7% accuracy. In [13], a CNN model was used to detect fire and smoke in the wild, consisting of different layers, such as convolutional, max-pooling, and dense layers, similar to those in VGG16 or MobileNet-V2. Two datasets were examined, one including offline images and the other consisting of images captured from different videos. The model achieved an accuracy of 95.41%. In [14], an FD model was proposed using Residual Networks (ResNet) to extract features and classify images using the Support Vector Machine (SVM). Two ensemble models were developed, achieving high classification accuracies of 98.91% and 99.15% utilizing a 10-fold cross-validation technique. In [15], an FD method was presented, using transfer learning to process a reduced dataset and pre-trained models, such as Xception, InceptionV3, and VGG16, to reduce computational complexity and maintain accuracy. Among the CNN models evaluated, the Xception model achieved the highest accuracy of 98.72% on two different datasets.

In [16], a hybrid model was proposed to detect fire intensity. This model combined Instance Segmentation (IS) and CNN, achieving a classification accuracy of 95.25%. The model also used IoT appliances to notify about fire severity. In [17], a deep-learning model was introduced to detect fires, incorporating image-preprocessing techniques for continuous FD and developed using Novel Dense Generative Adversarial Networks (NDGANs). Upon detection of fire or smoke, the system generated alarms, achieving an accuracy of 98.87%. In [18], a thermal imaging camera and seven FD sensors were used to compile fire data. The study involved training a CNN using thermal camera image data and Bidirectional Long Short-Term Memory (BiLSTM-Dense) for sensor data. The Densenet201 model demonstrated the highest accuracy of 0.99 for the image dataset, while the BiLSTM-Dense approach achieved an accuracy of 0.95 for the sensor dataset. However, 100% accuracy was achieved using a multimodal algorithm with the two datasets. The review in [19] offers an excellent resource to understand FD methods in greater detail.

Different CNN models have been proposed in the last decade. Each model's overall structure comprises input, convolutional, pooling, fully connected, and output layers. Each model has a unique topology that may vary in layer sizes, activation functions, and number of filters used. Well-known CNN models used in FD are AlexNet, GoogleNet (Inception-v3), ResNet, DenseNet, Visual Geometry Group (VGG), SqueezeNet [20], Xception [21], LeNet [22], and FireNet [23].

II. METHODOLOGY

A. Dataset

The dataset was downloaded from Kaggle [24] and comprises two folders containing fire and non-fire images. The first folder contains 755 images of fires, some of which feature heavy smoke from outdoor areas. The second collection features 244 natural images, including people, animals, trees, grass, waterfalls, lakes, rivers, foggy forests, and roads. Figure 1 shows a sample of images contained in the dataset.

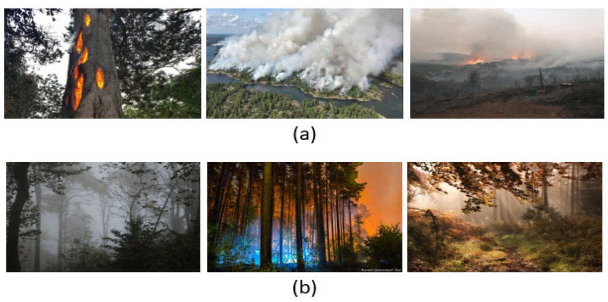

Fig. 1. Sample of dataset images: (a) fire images, (b) non-fire images.

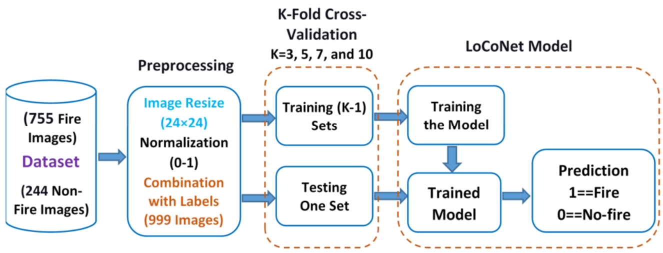

Fig. 2. Diagram of the proposed FD system.

B. Proposed FD System

The proposed FD system utilizes a CNN to detect fires in images. The CNN model extracts and classifies features from images accordingly, as shown in Figure 2.

1) Preprocessing

First, the images are loaded from two folders, one containing fire images and the other containing non-fire images. Each image is labeled with a binary number: 0 for non-existing fire and 1 for including fire. Then, all images are resized to the dimensions specified by the CNN type. After that, all the pixel values were divided by 255 to help the model converge quickly by maintaining the pixel values in a typical range between 0 and 1. Finally, both fire and non-fire images are merged into a dataset with their related labels.

2) K-Fold Cross-Validation

The dataset was divided into several groups using the K-fold cross-validation technique. The number of folds (*K*) was specified within the loop, such as 3, 5, 7, or 10. One fold was designated for the testing process, whereas the remaining *K*-1 folds were used to train the model. The test set was replaced with one of the training folds for each iteration to accurately evaluate the model's performance.

3) Constructing the CNN Model

The proposed LoCoNet model comprises six layers between the input and output layers. The layers are as follows.

a) Input Layer

The input layer sends the raw input data to the subsequent layer, without involving trainable parameters, such as biases or weights. The layer accepts color images (RGB) with a size of 24×24×3 pixels, chosen based on experimental results.

b) Convolutional and Max-Pooling Layers

The LoCoNet model comprises three convolutional layers after the input layer, each followed by a max-pooling layer. Each convolutional layer uses a 1×1 kernel size and the ReLU activation function. The first convolutional layer involves 32 filters to extract 32 feature maps from the input image with a size of 24×24. A max-pooling layer with a 2×2 pool size is then employed, so the feature dimensions are reduced by half without affecting their numbers. It conserves the most implied features from the last layer, where the output data size is 12×12×32. The second convolutional layer also comprises 32 filters of 1×1 kernel size applied to the previous layer's feature maps, which result in 32 features with the exact input size due to using a kernel size of 1×1. Once again, the second max-pooling layer reduces the spatial dimensions of the feature maps by half, resulting in 32 feature maps of size 6×6. Finally, the third combination of convolutional and max-pooling layers uses 32 filters of size 1×1, resulting in 3×3×32 output data size.

c) Flatten Layer

This layer converts the output of the convolutional and max-pooling layers to a 1-D vector before entering the fully connected layer. The size of features is converted from 3×3×32 to 288.

d) Fully Connected Dense Layer

After the flattened layer, a fully connected dense layer is added. This layer comprises 64 neurons, allowing it to learn high-level representations of the image features extracted by the convolutional layers. The sigmoid activation function is used in this layer, chosen for its suitability.

e) Dropout Layer

During training, this layer randomly drops 50% of its neurons to enhance the model's generalization and prevent overfitting. The data size remains unchanged at 64 as passed from the dense layer.

f) Output Layer

A single neuron is used in the output layer for classification. This neuron employs a sigmoid activation function to generate a probability score ranging from 0 to 1. If the score is above 0.5, the image is classified as fire; if not, it is classified as non-fire. It is worth noting that the LoCoNet model employs the Adam optimizer to minimize the loss.

4) Training and Evaluating the Model

After the dataset is divided into *K* sets for each fold, one set is used for validation, and the rest for training. The model is trained on the training data for specific epochs, such as 5, 10, 15, 20, 25, or 30. The learning rate is set to 0.001, and the batch size is 12. After training each fold, the model is evaluated on the validation set. The accuracy of each fold is recorded, and the average validation accuracy of all folds is calculated, resulting in the average accuracy for each epoch count.

III. EXPERIMENTAL RESULTS

The experiments were carried out on an MSI laptop running Windows with a 2.2 GHz processor and 16 GB of RAM. The proposed system was coded in Python and executed using the Visual Studio application. The proposed LoCoNet model processes color images with a size of 24×24 pixels. It is trained for different epochs (5, 10, 15, 20, 25, and 30) to examine the relationship between the number of epochs and accuracy. In addition, the model undergoes K-fold cross-validation for each epoch count, where *K* is selected as 3, 5, 7, and 10. The validation accuracy is calculated for each fold, and the average accuracy is determined for each epoch count. Table I presents the accuracy rates obtained from the LoCoNet model.

TABLE I. EXPERIMENTAL RESULTS OF THE LOCONET MODEL

No. of epochs	**Accuracy (%)**			
	3 Folds	***5 Folds***	***7 Folds***	***10 Folds***
5	95.10	96.00	95.80	96.80
10	98.00	97.30	97.10	96.20
15	99.00	98.40	97.10	96.89
20	99.00	99.00	98.40	97.30
25	97.40	99.00	97.10	98.40
30	99.00	98.50	97.00	97.40

The proposed LoCoNet model achieved an impressive accuracy of 99% using different epochs and folds. The lowest accuracy recorded was approximately 95.10%, which occurred at three folds and five epochs. The highest accuracy rate demonstrates the robustness of the proposed model.

A. Experiments Conducted with a Larger Dataset

It is highly recommended to evaluate the model on a larger dataset. Although the dataset includes 999 images, data augmentation can further expand its size. This process generates various versions of each image, resulting in a new dataset approximately five times larger than the original. The process was incorporated using TensorFlow's data augmentation layers. During training, the input images were revised to enhance inference capabilities. Data augmentation generates diverse shapes of fire and smoke to improve the reliability of the system. The details of the data augmentation layers used are presented as follows:

- Flipping: The input image is randomly flipped horizontally, vertically, or both. This technique helps the model identify fires or smoke with different orientations. The vertical flip reverses the image along the horizontal axis, whereas the horizontal flip reverses it along the vertical axis.
- Rotation: The image is randomly rotated for up to approximately ±72°, i.e., up to ±20% of 360°. This process makes the model invariant to rotational changes in input images, where the rotation portion is randomized within the specified range.
- Zooming: Up to 10% of the zoom level of the image is randomly modified, including both zooming in and out. This strategy models varying camera distances to detect fires and smoke at different scales more effectively.
- Translation: Each image is randomly shifted along the x and y axes by up to 10% of its width and height, respectively. This addresses situations where the object of interest is off-center, allowing the image content to be moved up, down, left, or right within the appointed limits.

All changes are applied to the images during training, so the augmented images are dynamically generated for each batch and not saved, which means that no additional memory is used. On the other hand, augmentation ensures that validation accuracy accurately reflects the model's performance on real-world unaltered images. This augmentation process benefits FD, making the model more resilient to environmental variations, such as camera angles, distances, or object positions.

The efficiency of the LoCoNet model in detecting fires and smoke was evaluated using sensitivity (*Sens*), specificity (*Spec*), accuracy (*Accu*), precision (*Prec*), and F1-score [14]:

$$Sens = \frac{TP}{TP+FN} \quad (1)$$

$$Spec = \frac{TN}{TN+FP} \quad (2)$$

$$Accu = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Prec = \frac{TP}{TP+FP} \quad (4)$$

$$F1-score = 2 \times \frac{Prec \times Sens}{Prec+Sens} \quad (5)$$

where TN refers to true negatives, TP to true positives, FN to false negatives, and FP to false positives. Table II presents the evaluation metrics after data augmentation.

TABLE II. PERFORMANCE METRICS USING A LARGE DATASET

Sens	*Spec*	*Accu*	*Prec*	F1-score
1.00	0.958333	0.99	0.987013	0.993464

Table II reveals the proposed model's reliability, where it achieved excellent results although the dataset was expanded to include different shapes and aspects of fires and smoke.

B. Comparative Results of Different CNN Models

The proposed LoCoNet model was compared with existing CNN models to assess its performance, advantages, and limitations. Table III compares various CNN models tested on the same platform and operating system. The training time presented reflects the duration measured during the training process, which utilized five folds and ten epochs for all models. The testing time indicates the duration of testing a single image from the dataset.

TABLE III. COMPARATIVE RESULTS OF DIFFERENT CNN MODELS

CNN model	No. of parameters	Memory usage (MB)	Training time (s)	Testing time (s)	Accuracy (%)
AlexNet	46,751,105	178.34	209.19	0.166	95.09
GoogleNet	140,253,317	535.02	207.02	0161	96.10
DenseNet201	18,691,013	71.30	719.75	4.99	98.20
SqueezeNet	2,207,813	8.42	189.44	0.332	94.89
Xception	21,648,685	82.58	134.46	0.984	94.49
LeNet5	183,725	0.7008	2.74	0.100	94.59
VGG19	26,317,381	100.39	380.73	0.299	96.60
FireNet-v2	6,861,413	26.17	49.90	0.162	96.60
LoCoNet	20,801	0.0793	2.86	0.102	99.00

Table III shows that the proposed LoCoNet model outperformed other CNN models in terms of accuracy and adaptability. In comparison, the DenseNet201 model reached an accuracy of 98.20%, close to that of the LoCoNet model (99%), but it requires significantly more processing time than the other models. Specifically, it takes about 719.75 seconds to train the model using 5-fold validation with ten epochs and 4.99 seconds to test a single image. This lengthy processing time is due to the large image size of 224×224×3 and the model's complexity, which comprises 18,691,013 parameters attributed to the increased number of layers and filters used. In contrast, the LeNet5 model has the shortest training time, approximately 2.74 seconds, but its accuracy is significantly lower than that of the LoCoNet model. The LeNet5 model uses 183,725 parameters and requires 0.7008 MB of memory, whereas the LoCoNet model uses only 20,801 parameters and 0.0793 MB (81.25 KB) of memory. Consequently, the LoCoNet model achieves the highest accuracy rate, consumes the least memory, and has a relatively low processing time. This efficiency is due to its use of a much smaller image size of 24×24×3, along with a minimal number of parameters, layers, and filters.

C. Comparison with the Related Approaches

It is essential to compare the results of the LoCoNet model with those of related works. Although researchers have introduced various approaches to address the FD problem, it is feasible to mention only some of them. Thus, the comparison focuses on the most recent and diverse methods that have achieved high accuracy in detecting fires. Numerous fire and smoke datasets are used in the literature, making it difficult to find approaches that employ DL with the same dataset. Consequently, the comparison is presented in Table IV without regard to the dataset type.

TABLE IV. COMPARISON WITH OTHER APPROACHES

Reference	CNN model	No. of epochs	*Accu* (%)
[12]	YOLOv3	200	83.70
[13]	Custom CNN	500	95.41
[14]	ResNet +SVM	-	99.15
[15]	Xception	75	98.72
[16]	IS+CNN	150	95.25
[17]	NDGANs	100	98.87
[18]	Densenet201	-	99.00
Proposed	LoCoNet	15	99.00

Table IV clearly shows that the accuracy of the proposed model is very similar to that achieved by other related methods, such as those of [14] and [18]. However, a key difference lies in the number of epochs used and the complexity of the models. In [14], the number of epochs was not specified, but the system's complexity is evident due to merging four ResNet models and using a further classifier, SVM, which increases processing time and memory usage. On the other hand, in [18], a maximum of 50 epochs was used, but it is unclear at which epoch the model achieved its highest accuracy. Additionally, this study employed the Densenet201 model, which is known as a complex and slow CNN model despite its robustness and reliability. Therefore, the proposed LoCoNet model has the lowest complexity because it utilizes a minimum image size of 24×24 pixels and a limited number of layers and filters. This strategy ensures a high level of accuracy while significantly decreasing processing time.

IV. CONCLUSION

This study introduces a low-complexity CNN model called LoCoNet, which was developed explicitly to detect fires in outdoor settings. Unlike other CNN models that utilize larger images, such as those sized at 224×224 pixels, LoCoNet operates with significantly smaller images sized at 24×24 pixels. Since the model is supervised, it does not require the same level of detail in each image pixel as for tasks such as face recognition. Thus, this minimized image size is sufficient to determine the presence of fire. Additionally, the LoCoNet model comprises three convolutional layers, each with 32 filters, with a minimum kernel size of 1×1, whereas the fully connected layer contains 64 filters. The experiments showed that the LoCoNet model achieved an impressive accuracy of approximately 99% on both the original dataset and its expansion. This achievement positions the proposed model as superior to previous CNN models, with the closest competitor being Densenet201, which achieved an accuracy of approximately 98.10%. However, Densenet201 is much more complex than the LoCoNet model. When comparing the LoCoNet model to related approaches, it is evident that its accuracy is nearly on par with the best-performing methods. The simplicity of the LoCoNet model provides an advantage over others in terms of complexity. Future research could explore its application in various fields, including online FD.

REFERENCES

[1] C. H. Lee, W. H. Lee, and S. M. Kim, "Development of IoT-Based Real-Time Fire Detection System Using Raspberry Pi and Fisheye Camera," *Applied Sciences*, vol. 13, no. 15, Jan. 2023, Art. no. 8568, https://doi.org/10.3390/app13158568.

[2] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, https://doi.org/10.1007/s00521-023-08809-1.

[3] Z. S. A. Hakeem, H. I. Shahadi, and H. H. Abass, "An Automatic System for Detection of Fires in Outdoor Areas," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Prague, Czech Republic, Jul. 2022, pp. 1–6, https://doi.org/10.1109/ICECET55527.2022.9872883.

[4] F. Khan, Z. Xu, J. Sun, F. M. Khan, A. Ahmed, and Y. Zhao, "Recent Advances in Sensors for Fire Detection," *Sensors*, vol. 22, no. 9, Jan. 2022, Art. no. 3310, https://doi.org/10.3390/s22093310.

[5] Z. S. A. Hakeem, H. I. Shahadi, and H. H. Abbas, "An automatic flame detection system for outdoor areas," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 4, pp. 864–871, Aug. 2023, https://doi.org/10.12928/telkomnika.v21i4.24381.

[6] G. V. Kuznetsov, R. S. Volkov, A. S. Sviridenko, and P. A. Strizhak, "Reduction of response time of fire detection and containment systems in compartments," *Fire Safety Journal*, vol. 144, Mar. 2024, Art. no. 104089, https://doi.org/10.1016/j.firesaf.2024.104089.

[7] P. V. B. Ngoc, L. H. Hoang, L. M. Hieu, N. H. Nguyen, N. L. Thien, and V. T. Doan, "Real-Time Fire and Smoke Detection for Trajectory Planning and Navigation of a Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11843–11849, Oct. 2023, https://doi.org/10.48084/etasr.6252.

[8] K. Kikuta, K. T. Murata, and Y. Murakami, "A Daytime Smoke Detection Method Based on Variances of Optical Flow and Characteristics of HSV Color on Footage from Outdoor Camera in Urban City," *Fire Technology*, vol. 60, no. 3, pp. 1427–1452, May 2024, https://doi.org/10.1007/s10694-023-01522-4.

[9] F. Gong *et al.*, "A Real-Time Fire Detection Method from Video with Multifeature Fusion," *Computational Intelligence and Neuroscience*, vol. 2019, no. 1, 2019, Art. no. 1939171, https://doi.org/10.1155/2019/1939171.

[10] A. Hosseini, M. Hashemzadeh, and N. Farajzadeh, "UFS-Net: A unified flame and smoke detection method for early detection of fire in video surveillance applications using CNNs," *Journal of Computational Science*, vol. 61, May 2022, Art. no. 101638, https://doi.org/10.1016/j.jocs.2022.101638.

[11] H. R. Farhan, M. H. Al-Muifraje, and T. R. Saeed, "Face recognition system based on continuous one-state model," *AIP Conference Proceedings*, vol. 2144, no. 1, Aug. 2019, Art. no. 050001, https://doi.org/10.1063/1.5123117.

[12] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Studies in Thermal Engineering*, vol. 19, Jun. 2020, Art. no. 100625, https://doi.org/10.1016/j.csite.2020.100625.

[13] J. S. Almeida, S. K. Jagatheesaperumal, F. G. Nogueira, and V. H. C. de Albuquerque, "EdgeFireSmoke++: A novel lightweight algorithm for real-time forest fire detection and visualization using internet of things-human machine interface," *Expert Systems with Applications*, vol. 221, Jul. 2023, Art. no. 119747, https://doi.org/10.1016/j.eswa.2023.119747.

[14] S. Dogan *et al.*, "Automated accurate fire detection system using ensemble pretrained residual network," *Expert Systems with Applications*, vol. 203, Oct. 2022, Art. no. 117407, https://doi.org/10.1016/j.eswa.2022.117407.

[15] V. E. Sathishkumar, J. Cho, M. Subramanian, and O. S. Naren, "Forest fire and smoke detection using deep learning-based learning without forgetting," *Fire Ecology*, vol. 19, no. 1, Feb. 2023, Art. no. 9, https://doi.org/10.1186/s42408-022-00165-0.

[16] S. J. Malebary, "Early Fire Detection Using Long Short-Term Memory-Based Instance Segmentation and Internet of Things for Disaster Management," *Sensors*, vol. 23, no. 22, Jan. 2023, Art. no. 9043, https://doi.org/10.3390/s23229043.

[17] T. Shawly and A. A. Alsheikhy, "Fire Identification Based on Novel Dense Generative Adversarial Networks," *Artificial Intelligence Review*, vol. 57, no. 8, Jul. 2024, Art. no. 207, https://doi.org/10.1007/s10462-024-10848-6.

[18] A. Sharma *et al.*, "Fire Detection in Urban Areas Using Multimodal Data and Federated Learning," *Fire*, vol. 7, no. 4, Apr. 2024, Art. no. 104, https://doi.org/10.3390/fire7040104.

[19] G. Cheng, X. Chen, C. Wang, X. Li, B. Xian, and H. Yu, "Visual fire detection using deep learning: A survey," *Neurocomputing*, vol. 596, Sep. 2024, Art. no. 127975, https://doi.org/10.1016/j.neucom.2024.127975.

[20] J. Gotthans, T. Gotthans, and R. Marsalek, "Deep Convolutional Neural Network for Fire Detection," in *2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA)*, Bratislava, Slovakia, Apr. 2020, pp. 1–6, https://doi.org/10.1109/RADIOELEKTRONIKA49387.2020.9092344.

[21] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and Md. M. Rahman, "An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease diagnosis," *Healthcare Analytics*, vol. 3, Nov. 2023, Art. no. 100143, https://doi.org/10.1016/j.health.2023.100143.

[22] R. A. Hazarika, A. Abraham, D. Kandar, and A. K. Maji, "An Improved LeNet-Deep Neural Network Model for Alzheimer's Disease Classification Using Brain Magnetic Resonance Images," *IEEE Access*, vol. 9, pp. 161194–161207, 2021, https://doi.org/10.1109/ACCESS.2021.3131741.

[23] A. Shees, M. S. Ansari, A. Varshney, M. N. Asghar, and N. Kanwal, "FireNet-v2: Improved Lightweight Fire Detection Model for Real-Time IoT Applications," *Procedia Computer Science*, vol. 218, pp. 2233–2242, 2023, https://doi.org/10.1016/j.procs.2023.01.199.

[24] A. Saied, "FIRE Dataset." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/phylake1337/fire-dataset.