

# Deep Learning-Based Fire Detection for Enhanced Safety Systems

Mothefer Majeed Jahefer<sup>1</sup><sup>\*</sup>

<sup>1</sup>First Author Affiliation,

Department of Computer Techniques Engineering, Imam Al-Kadhum College (IKC), Baghdad, Iraq

\*Corresponding Author: Mothefer Majeed Jahefer

DOI: <https://doi.org/10.31185/wjps.221>

Received 11 August 2023; Accepted 24 September 2023; Available online 30 December 2023

**ABSTRACT:** Fire detection systems are a critical aspect of modern safety and security infrastructure, playing a pivotal role in safeguarding lives and property against the destructive force of fires. Rapid and accurate identification of fire incidents is essential for timely response and mitigation efforts. Traditional fire detection methods have seen substantial advancements, but with the advent of computer vision technologies, the field has undergone a transformative shift. This paper presents a method for fire detection using deep convolutional neural network (CNN) models. This approach employs transfer learning, utilizing two pre-trained CNN models from the ImageNet dataset: VGG (Visual Geometry Group) and InceptionV3, to extract valuable features from input images. These extracted features serve as input for a machine learning (ML) classifier, namely the Softmax classifier. The Softmax activation function computes the probability distribution to assign accurate class probabilities for discriminating between two types of images: fire and non-fire. Experimental results demonstrate VGG16 accuracy of 96.57%, Recall of 96.84%, Precision of 96.33%, and an F1 score of 96.58%. In comparison, Inception-v3 achieved an accuracy of 97.89%, Recall of 97.89%, Precision of 97.89%, and an F1 score of 97.89%. These results highlight that the proposed method successfully detects fire areas and achieves seamless classification performance compared to other current fire detection methods.

**Keywords:** Fire detection, Deep learning, CNN, VGG16, inceptionV3



## 1. INTRODUCTION

Fire outbreaks pose substantial risks to human lives, property, and the environment, leading to immense tolls on both natural and human fronts. Fires, whether ignited by natural causes or human activities, can escalate swiftly, engulfing structures and landscapes in a matter of minutes. Traditionally, fire detection has relied on a range of methods including manual surveillance and sensors [1]. These sensors can measure temperature, gases, and smoke [2]. However, these methods often suffer from limitations such as delays in detection, false alarms, and challenges in handling complex fire scenarios. They may have been successful for small fires and confined spaces, but are ineffective for larger fires that can grow rapidly, consuming entire regions with disastrous effects [3].

Swift and precise fire detection holds utmost importance in facilitating prompt emergency responses, mitigating potential losses, and upholding public safety. The emergence of deep learning techniques offers a new avenue to address these challenges by leveraging the power of artificial intelligence to analyze and interpret visual data for real-time fire detection. With the advent of IoT-enabled smart cities, it becomes increasingly important to ensure the safety of citizens and their properties [4]. Deep learning for detection and recognition is an integral part of computer vision technologies [5]. The strength of using video in fire detection lies in its ability to monitor large and open spaces. Current fire and flame detection methods rely on the use of color and motion information in video [6]. While various deep learning models like neural networks, recurrent neural networks, etc., have been explored, deep convolutional neural networks (CNNs) [7] are particularly adept at finding patterns in images. In this paper, we elucidate the significance of two distinguished models, namely Inception and VGG16, which have emerged as formidable assets in the domain of fire detection. Inception,

characterized by its InceptionV3 architecture, and VGG16, celebrated for its VGG Net architecture, have consistently demonstrated exceptional proficiency across various computer vision tasks. Their remarkable ability to capture intricate patterns and hierarchical features within images positions them as highly promising contenders for the task of fire detection. This introduction embarks on an exploration of how the integration of Inception and VGG16 models is revolutionizing the landscape of fire detection, fundamentally transforming our approaches to recognizing and responding to fire emergencies. In the forthcoming sections, we will delve into the methodologies employed to leverage the feature extraction capabilities of these models, their practical applications in real-time fire detection scenarios, and the profound impact they wield in enhancing the precision and efficiency of fire recognition systems. Furthermore, we will delve into the pivotal role these models play in fortifying safety measures and bolstering disaster management strategies, ultimately contributing to the safeguarding of lives and valuable assets across a wide array of contexts where fire detection stands as a paramount concern. The main contributions of this paper include:

- Examining two deep CNN models, VGG16 and InceptionV3, for fire detection in surveillance cameras.
- Leveraging TL to address data scarcity, enhance feature representation, and reduce the risk of overfitting.
- Employing augmentation techniques to artificially expand the dataset's size and diversity by applying various transformations to the original data.
- The Inception model achieved an accuracy of 97.89%, while the VGG16 model achieved an accuracy of 96.57%. These results demonstrate superior performance compared to the best existing methods.

## 2. RELEATED WORK

An increasing fascination has arisen surrounding the application of deep learning techniques for fire detection within smart urban environments. A method rooted in YOLOv3 architecture was put forth to detect fires in outdoor settings. This methodology was tested on a collection of real-world images [8]. In the work of [9], they introduced a fire detection system (FFire Net) that leveraged the MobileNetV2 model for distinguishing forest fires. In [10], a hybrid approach was presented, which combined long short-term memory (LSTM) with the You Only Look Once (YOLO) framework to identify smoke in wildfire conditions. The authors employed a streamlined teacher-student LSTM architecture within their proposed solution, effectively reducing the layer count while achieving enhanced smoke detection outcomes. In reference [11], an innovative CNN design focused on energy efficiency and computational efficiency was put forth, drawing inspiration from the Squeeze Net architecture. This design was intended for tasks such as fire detection, localization, and interpreting the visual characteristics of fires as captured by CCTV surveillance networks. Frizzi [12] introduced a nine-layer convolutional neural network designed to identify instances of fire or smoke within videos. Hohberg [13] undertook training of a convolutional neural network to accurately recognize wildfire smoke. In the context of [14], a pioneering system for early detection of wildfire smoke was put forward. This system utilized an upgraded variant of YOLOv5 on images acquired through unmanned aerial vehicles (UAVs). In reference [15], the detection of color variations in fire is accomplished by performing spatial wavelet transforms on regions with moving fire-colored elements. In other research, they employ the RGB model, which examines red, green, and blue colors in videos, to differentiate fire-like and smoke-like pixels. Pixel identification as fire involves assessing red color intensity and brightness, with criteria including evaluating whether the pixel resembles a developing flame and exhibits chaotic characteristics akin to real fires [16]. In reference [17], highly effective fire detection algorithms using advanced CNN models, specifically YOLOv3, were introduced. This advancement is pivotal for detecting fires early on, leading to substantial reductions in potential losses. In reference [18], the author investigates the utilization of AI-powered computer vision techniques for fire and smoke detection in images. It employs the Faster R-CNN architecture as a component of Convolutional Neural Networks (CNNs). In reference [19], the study presents a time-effective CNN for fire detection in surveillance systems, leveraging Deep ANN and Alex Net through transfer learning. The model's effectiveness is confirmed in detecting fires in surveillance videos and benchmarked against cutting-edge models.

## 3. TRANSFER LEARNING APPROCH

A transfer learning approach is a machine learning technique where a model developed for a particular task is reused or adapted as the starting point for a model on a second task. In transfer learning, knowledge learned from one problem domain is applied to a related but different problem domain. This technique is especially valuable when you have limited data for the target task or when you want to improve the performance of a model on a new task.

Here's a breakdown of what transfer learning typically handles:

1. **Pre-trained Model:** Transfer learning starts with a model pre-trained on a large dataset for a related task. These pre-trained models are often trained on extensive and diverse datasets, like ImageNet for image classification or large text corpora for natural language understanding. These models have already learned to extract useful features or representations from data.
2. **Feature Extraction:** Instead of building a model from scratch, you use the pre-trained model's layers as a feature extractor. These layers have already learned to recognize general patterns and features within the data. You can take the output from these layers as a fixed set of features or representations for your new task. This step is particularly useful in computer vision and natural language processing tasks.
3. **Fine-Tuning:** In many cases, you might also fine-tune some of the later layers of the pre-trained model on your specific task. This means that you allow some of the model's parameters to be updated or adapted to the new task while keeping the knowledge gained from the original task intact. This can help the model specialize for your particular problem.
4. **Transfer of Knowledge:** The primary goal of transfer learning is to leverage the knowledge learned in the source task to improve the performance of the model on the target task. This can result in faster training and better generalization, especially when you have limited data for the target task.
5. **Domains:** Transfer learning can be applied in various domains, including computer vision, natural language processing, and speech recognition. For example, a model trained to classify animals in images might be adapted to classify vehicles with relatively little additional training.

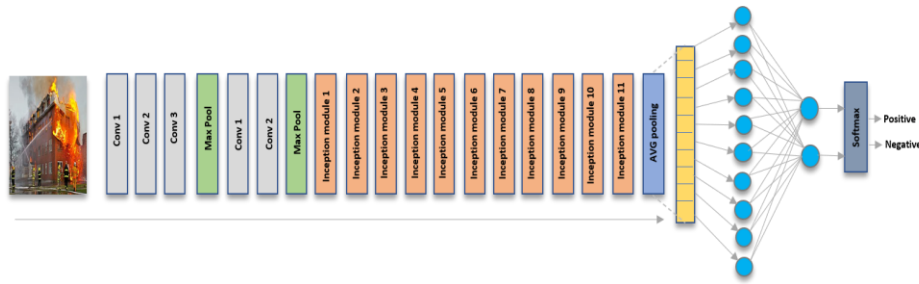
Overall, transfer learning is a powerful approach in machine learning because it allows you to build effective models for new tasks with less data and computational resources, while benefiting from the knowledge and representations learned in previous tasks.

### 3.1 CNN models

#### 3.2.1 Inception-v3 model

Inception-v3 is a deep convolutional neural network developed by Google specifically for image classification tasks in computer vision. It stands out for its innovative Inception modules, which employ parallel convolutional operations with varying filter sizes to capture features across different scales. This architecture also incorporates factorization techniques, reducing the number of parameters and enhancing overall efficiency.

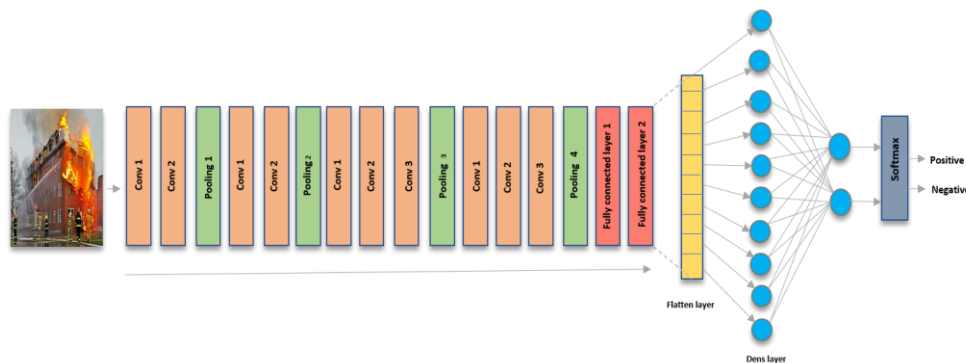
Instead of fully connected layers, Inception-v3 employs Global Average Pooling (GAP) and integrates auxiliary classifiers during training to aid in convergence. It has achieved state-of-the-art performance on the ImageNet dataset and finds wide application in tasks such as image recognition and object detection. Inception-v3 serves as a foundational model for transfer learning and is part of the Inception family, including successors like Inception-v4 and Inception-Res Net models. The architecture of Inception-v3 comprises distinct components: the stem, Inception blocks, and concluding layers [20]. The stem's purpose is to reduce the spatial resolution of the input image, often achieved through a series of convolutional layers. This reduction helps lower computational demands in subsequent layers of the network. Central to the architecture are the Inception blocks, which play a pivotal role in increasing the network's depth. Each Inception block combines convolutional layers with Inception modules. The Inception modules contribute to capturing features across various scales, while the convolutional layers enhance the network's overall depth. Towards the network's culmination, the final layers of Inception-v3 further reduce the spatial dimensions of feature maps, ultimately producing the conclusive output. These closing layers typically consist of several convolutional layers coupled with a global average pooling layer. The output from these final layers is then processed through a fully connected layer, leading to the ultimate classification [21] [22]. For a visual representation of the Inception-v3 model's architecture, please refer to Figure 1.



**Fig 1.** The Architecture of InceptionV3

### 3.2.2. VGG16 model

The VGG16 model, a key member of the VGG family of convolutional neural networks, was introduced by Simonyan and Zisserman following the impressive success of CNNs in image recognition [23]. This architecture is known for its straightforward yet highly effective framework in constructing deep neural networks. Instead of using larger filters like 5x5 or 11x11, VGG16 adopts a unique approach of stacking layers of 3x3 filters. This innovative strategy demonstrated that using these smaller filters in parallel could yield comparable results to larger ones, while also reducing computational complexity by minimizing the number of parameters. Additionally, VGG16 incorporates 1x1 convolutions within the convolutional layers, which help control network complexity. These 1x1 convolutions enable the learning of linear combinations of subsequent feature maps. To maintain spatial resolution, a max pooling layer with padding is added after each convolutional layer. This combined methodology contributes to VGG's reputation for increased depth, consistent topology, and simplicity. The VGG Networks introduced various network architectures, with VGG16 being a noteworthy example. VGG16 consists of 16 weight layers and follows a similar structural pattern as other models in the VGG series, involving a sequence of convolutional layers followed by fully connected layers. For a visual representation of the VGG16 architecture, refer to Figure 2.



**Fig 2.** The Architecture of VGG16

### 3.3. Dataset

In our fire detection research, we utilized a dataset obtained from Kaggle [24], encompassing a wide range of fire-related scenarios, including outdoor wildfires and industrial fires. The dataset also covers various environmental conditions, allowing us to thoroughly evaluate the robustness and adaptability of our fire detection algorithms. This dataset sourced from Kaggle underwent meticulous vetting to ensure high quality and precision in annotations, providing a solid foundation for our model training process. Additionally, the dataset is substantial, containing a significant number of samples that enable comprehensive training. Its ample size empowers our models to learn and generalize effectively, ultimately enhancing their proficiency in accurately detecting fires in real-world settings. For our training purposes, we categorized the images into two distinct groups: one depicting fire and the other representing non-fire scenes, as illustrated in figures 3 and 4. A total of 2,641 images were used for training, while an additional set of 620 images was exclusively reserved for testing. To ensure seamless compatibility between the dataset and our proposed model, we carefully resized the images to specific dimensions of  $224 \times 224 \times 3$ , representing height, width, and the number of color channels (RGB). This resizing process was crucial to align the images with the prescribed input size requirements of our selected deep learning models, InceptionV3 and VGG16. Recognizing the critical role of data availability and its direct impact on model performance, we implemented data augmentation. This technique involved generating modified versions of the



existing data, effectively expanding our training dataset artificially. Data augmentation served the dual purpose of addressing potential limitations stemming from a finite dataset and empowering our model with enhanced adaptability and resilience.

The augmentation process involved various transformations, including image rotations, horizontal and vertical flips, shifts in image position, and scaling operations. By applying these diverse modifications to the original images, we aimed to simulate different perspectives, orientations, and scales that may be encountered in real-world fire incidents.

In essence, data augmentation served as a critical tool to enhance the model's ability to generalize beyond the confines of the training dataset. It enabled the model to learn a broader spectrum of features and patterns, ultimately resulting in a more robust and accurate fire detection system when faced with new, previously unseen fire-related images.



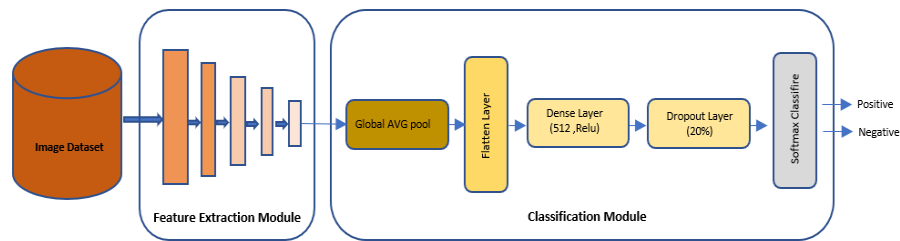
**Fig 3.** Fire Dataset



**Fig 4.** Non-Fire Dataset

## 4. THE PROPOSED MODEL

In this study, we employed the intelligent technique of transfer learning, which utilizes the robust architectures of InceptionV3 and vgg16. This process involved two main steps: initially, the model was pre-trained on a large dataset, and then it underwent fine-tuning to customize it for our specific task. This approach enabled us to extract meaningful features from our input data, laying the groundwork for a novel classifier. For a visual representation of our methodology and system framework, Take a look at Figure 2. Our system primarily relies on the transfer learning capabilities of InceptionV3 and vgg16, both known for their effectiveness in image classification. It is specifically designed to discern whether a given image depicts a fire or not, thereby classifying the parameters associated with fire status:



**Fig 5.** schematic diagram of the proposed model

### A- Feature Extraction Module

InceptionV3 and VGG16 models refer to a key component in machine learning and computer vision pipelines that extracts meaningful and informative features from input data, typically images. While the general concept is the same for both models, there are some differences in how they achieve feature extraction:

1. **Pre-trained Models:** In both cases, you start with a pre-trained deep learning model. These models have already learned to recognize various patterns, shapes, and features from a large dataset, typically ImageNet, which makes them valuable for feature extraction.
2. **Removing Classification Layers:** The first step in creating a feature extraction module is to remove the final classification layers of these pre-trained models. These layers are responsible for predicting specific categories or labels and are usually not needed for feature extraction.
3. **Freezing Model Weights:** To ensure that the knowledge gained during the pre-training is retained, you freeze the weights of the layers in the model. This means that during feature extraction, the model's weights won't be updated based on your specific dataset.
4. **Choosing a Feature Layer:** Both InceptionV3 and VGG16 models consist of multiple layers, each capturing different levels of abstraction. You must select a layer from the model that provides the level of detail or abstraction suitable for your particular task. For instance, earlier layers capture low-level features like edges, while deeper layers capture more complex patterns.
5. **Feature Extraction:** Input data (usually images) are passed through the pre-trained model until the chosen feature extraction layer. This process transforms the input data into a set of feature vectors, each representing various aspects of the input.
6. **Utilizing Extracted Features:** The extracted features can then be used for various
7. downstream tasks such as image classification, object detection, style transfer, or any application where having rich and informative feature representations is advantageous.

However, there is a distinct contrast among the models outlined in Section A, specifically in relation to the Feature Extraction Module:

1. **Architecture:** InceptionV3 and VGG16 models are two different convolutional neural network (CNN) architectures. Inception models are known for their efficiency in capturing diverse feature scales, while VGG16 models have a simple and uniform architecture that is effective for feature extraction.
2. **Layer Choice:** The choice of the specific layer for feature extraction may differ between the two models. InceptionV3 models often have layers with various filter sizes, enabling the capture of features at multiple scales. VGG16, on the other hand, has a more uniform structure, and you typically choose layers deeper in the network for higher-level features.
3. **Output Size:** The size and dimensionality of the extracted features may vary between the two models. InceptionV3 models might provide more complex and multi-scale features due to their architecture, while VGG16 might offer features with a consistent spatial resolution.

In summary, a Feature Extraction Module for both InceptionV3 and VGG16 models is a critical component in various computer vision tasks. It leverages the power of pre-trained deep learning models to extract informative features from input data, enabling better performance in subsequent tasks without the need for training a model from scratch.

### B- Classification Module

The classification module is tasked with refining the weights of characteristic vectors to discern individuals with injuries. This module is constructed with the sequential layers

1. **Global Average Pooling Layer** use to condense the spatial dimensions of input feature maps into a fixed-length feature vector, a global average pooling layer is employed. The calculation is governed by the formula [25]:

$$G_i = \frac{1}{H \times w} \times \sum \sum f(i, j, k) \quad (1)$$

In equation (1), the summation is carried out over all spatial dimensions of the feature map, where 'j' spans from 1 to 'H' and 'k' spans from 1 to 'W'. With a feature map denoted as 'F' possessing dimensions of H×W×C (height, width, and number of channels), the outcome of the global average pooling layer, termed 'G', materializes as a C-dimensional vector. Within this vector, each element 'i' signifies the average of all activations in the feature map 'F', encompassing all spatial dimensions.

2. **Flatten Layer** This layer converts the output from the prior layer into a 512-feature vector, suitable for input into a subsequent fully connected layer. Also,
3. **Dense (Fully Connected) Layer** this comprising 512 neurons, this layer employs a Rectified Linear Unit (ReLU) activation function. Each neuron establishes connections with 512 nodes from the feature vector.
4. **Dropout Layer** use to counteract overfitting and augment the model's capacity to handle noise while generalizing to novel data, a dropout layer is introduced. A dropout rate of 0.2 is applied, indicating that during training, 20% of the neurons are randomly deactivated. And last is
5. **Softmax Layer** is conclusive step entails employing a Softmax layer to classify features and establish the ultimate probabilities for the two classes within the dataset[17]. The Softmax activation function is defined as follows [26]:

$$\text{softmax}(c_i) = \frac{e^{c_i}}{\sum_{j=1}^k e^{c_j}} \quad (2)$$

This orchestrated sequence of layers within the classification module enables the model to effectively differentiate and categorize the two types of classes present in the dataset

### C- Metrics for Performance Assessment

To gauge the proficiency of the introduced system, its effectiveness is assessed through a range of metrics that vividly showcase its capabilities. These evaluation criteria encompass accuracy, precision, recall, and F1-score. The subsequent explanations outline the definitions of these metrics [5]:

- **Accuracy:** Serving as a pivotal yardstick, accuracy quantifies the system's ability to correctly classify both normal and anomalous instances, considering the entire dataset. This metric particularly proves useful when the dataset maintains a balanced distribution of instances (Equation 3).

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

- **Precision:** Precision captures the ratio of accurately classified positive instances to the total instances classified as positive. It delineates the model's capacity to minimize false positives, which is essential in scenarios where misclassification is costly (Equation 4).

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

- **Recall:** Recall signifies the ratio of accurately classified positive instances to the actual total positive instances present in the dataset. It is a crucial measure for situations where missing true positive instances carries significant consequences (Equation 5).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

- **F1-score:** An amalgamation of precision and recall, the F1-score is employed to strike a balance between these two metrics. It conveys the harmonic mean between precision and recall, encapsulating a comprehensive assessment of the system's performance (Equation 6).

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

These performance evaluation metrics collectively shed light on various aspects of the system's effectiveness, revealing its prowess in handling both fire and no fire instances within the dataset.

#### 4.1 Experimental Configuration and Training Parameters

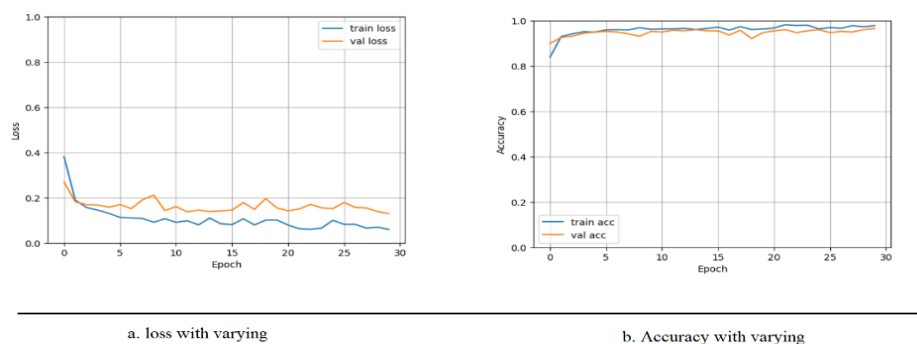
The experimentation process was executed using the Python programming language, employing the Keras library in conjunction with the TensorFlow platform. Keras, a Python-based high-level API for neural networks, was harnessed atop the TensorFlow framework. The computational setup employed for both training and testing the system featured the following specifications:

- CPU: 12th Generation Intel Core i7-1265H (2.30 GHz, 10 cores)
- GPU: NVIDIA GeForce RTX 3070 (8 GB)

These meticulously selected configurations contributed to the robustness and effectiveness of the training process, leading to a proficient model capable of capturing essential patterns in the data.

### 5 RESULTS

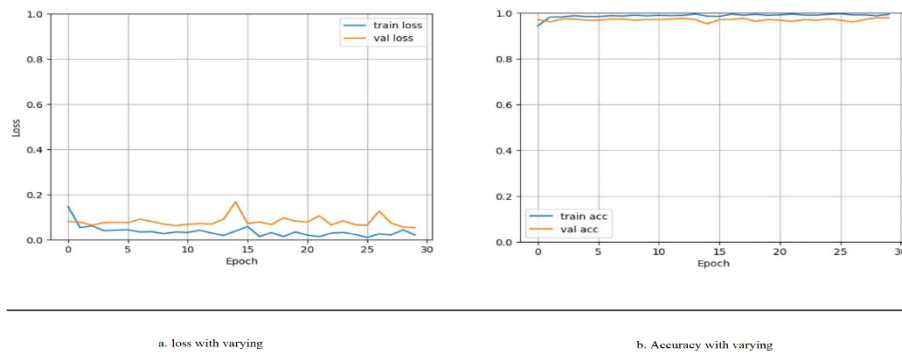
In the context of this study, we undertook a comprehensive evaluation of the effectiveness of our proposed methodology in identifying fire-related scenarios. This assessment was carried out by employing two prominent convolutional neural network (CNN) architectures: VGG16 and InceptionV3. Both architectures were rigorously tested across our dataset to gauge their performance. The results of the training and validation phases, encompassing metrics such as losses, accuracies, have been visually represented in



**Fig 6 .Experimental Results using VGG16 Model**

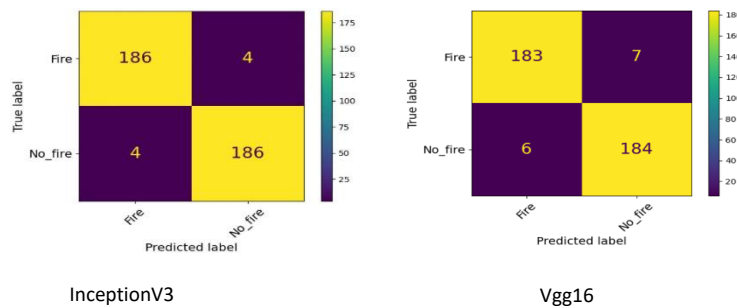
Figures 6 and 7. These graphical representations provide a clear and vivid depiction of how the models fared in terms of their ability to discern fire situations. Through these visual aids, we gain a comprehensive understanding of the models' capabilities and insights into the dynamics governing their functionality.





**Fig 7.** Experimental Results using InceptionV3 Model

Also, when comparing the confusion matrices of the VGG16 and InceptionV3 models, it becomes evident that the InceptionV3 model yields superior outcomes. The analysis of the confusion matrices suggests that the InceptionV3 model outperforms the VGG16 model in terms of accuracy and precision. This improvement in performance is particularly noticeable in correctly identifying and categorizing various classes within the dataset. Consequently, based on the assessment of the confusion matrices, it can be concluded that the InceptionV3 model is more effective and reliable in its predictions compared to the VGG16 model see figure 8.



**Fig 8.** confusion matrices

Achievement across Training and Validation The models' accomplishments throughout the training and validation stages are vividly captured. The comprehensive assessment of these models' performances on the dataset is meticulously documented in Table 1. This assessment presents a thorough exposition of the model's achievements in the domains of accuracy, recall, precision, and F1 score.

Table 1. The Experimental Results on dataset Using VGG16 and Inception-v3 models

model	Accuracy %	Recall %	Precision %	F1 score %
VGG16	96.57	96.84	96.33	96.58
Inception-v3	97.89	97.89	97.89	97.89

## 6 DISCUSSION

In this study, we conducted experiments using both the VGG16 and InceptionV3 networks for feature extraction. The initial experiment with the VGG16 network resulted in an impressive accuracy of 96.57%. This highlights the model's ability to accurately classify a substantial portion of the input images. The recall metric, indicating the model's precision in identifying positive instances, achieved a commendable score of 96.84%. Furthermore, the precision metric, reflecting the model's accuracy in labeling instances as positive, was recorded at 96.33%. The F1 score, which provides a balanced assessment of both precision and recall, reached a noteworthy value of 96.58%. In the subsequent

experiment using the InceptionV3 network, we achieved an even higher accuracy of 97.89%, slightly surpassing the performance of the VGG16 model. However, the recall metric showed a value of 97.89%, indicating that the InceptionV3 model may have a relatively reduced capacity to accurately identify positive instances compared to the VGG16 model. On the other hand, the precision metric demonstrated an elevated value of 97.89%, suggesting that the InceptionV3 model excelled in accurately assigning positive labels compared to the VGG16 model. The F1 score, at 97.89%, highlights the harmonious balance between precision and recall within the Inception-v3 model. Our primary aim was to thoroughly evaluate two prominent convolutional neural network architectures, namely VGG16 and InceptionV3, with a focus on their effectiveness in feature extraction for fire situation detection. The findings from our investigation provide crucial insights into the capabilities of these architectures. Notably, the VGG16 network displayed commendable performance, achieving an accuracy of 96.57%. This result underscores its proficiency in accurate classification. nevertheless, the Inception-v3 network demonstrated a slightly higher accuracy of 97.89%, indicating its potential for heightened discernment. It is important to note a discernible trade-off between recall and precision metrics. While the InceptionV3 model demonstrated superior precision, it also exhibited relatively lower recall compared to the VGG16 model. This implies its inclination towards correctly labeling positive instances, albeit with a likelihood of some instances going unnoticed. Furthermore, after inspecting our outcomes alongside those of models proposed by various authors (as outlined in Table 2), it is evident that our approach has demonstrated notably competitive performance. Specifically, the fine-tuned Inception-v3 model has achieved a level of accuracy that surpasses the results reported in [30] and outperforms other methods. This highlights its notably superior performance, signifying its effectiveness in the given context.

Ref. year	model	Accuracy %
2020 [17]	YOLO v3	83.7
2021 [18]	Faster R-CNN	80
2022 [19]	Deep ANN and Alex Net	95 and 98
proposed model	VGG16	96.57
proposed model	Inception-v3	97.89

## 7 CONCLUSIONS

In conclusion, both architectural models exhibit promising outcomes, each with its distinct strengths. InceptionV3 excels notably in precision, demonstrating a keen ability to accurately identify positive instances. Conversely, VGG16 maintains a balanced performance, showcasing proficiency across various metrics. The selection between these models should be contingent upon the specific requirements of the application, with careful consideration of the relative importance of precision and recall in accordance with the desired outcomes. This evaluation provides a solid foundation for informed decision-making in the context of fire situation detection.

## REFERENCES

- [1] F. Zhang, P. Zhao, S. Xu, Y. Wu, X. Yang, and Y. Zhang, "Integrating multiple factors to optimize watchtower deployment for wildfire detection," *Sci. Total Environ.*, vol. 737, p. 139561, 2020.
- [2] F. Zhang, P. Zhao, J. Thiyaalingam, and T. Kirubarajan, "Terrain-influenced incremental watchtower expansion for wildfire detection," *Sci. Total Environ.*, vol. 654, pp. 164–176, 2019.
- [3] F. Carta, C. Zidda, M. Putzu, D. Loru, M. Anedda, and D. Giusto, "Advancements in forest fire prevention: A comprehensive survey," *Sensors*, vol. 23, no. 14, p. 6635, 2023.
- [4] A. Gohari, A. Bin Ahmad, R. B. A. Rahim, A. S. M. Supa'at, S. Abd Razak, and M. S. M. Gismalla, "Involvement of surveillance drones in smart cities: A systematic review," *IEEE Access*, vol. 10, pp. 56611–56628, 2022.
- [5] S. A. Jebur, K. A. Hussein, H. K. Hoomod, L. Alzubaidi, & J. Santamaría (2022). Review on deep learning approaches for anomaly event detection in video surveillance. *Electronics*, 12(1), 29.
- [6] M. Taha, & S. Alsaidi (2022). A Review of Classifications Techniques and Computer Aided used for Breast Cancer Detection. *Wasit Journal of Pure Sciences*, 1(2), 260-271.
- [7] L. R. A. Al-Khazraji, A. R. Abbas, & A. S. Jamil (2023). A Systematic Review of Deep Dream. *IRAQI J. Comput. Commun. Control Syst. Eng.*, 23, 192-209.
- [8] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Mini-YOLOv3: real-time object detector for embedded applications," *IEEE Access*, vol. 7, pp. 133529–133538, 2019.

- [9] S. Khan and A. Khan, "Ffirenet: Deep learning based forest fire classification and detection in smart cities," *Symmetry (Basel)*, vol. 14, no. 10, p. 2155, 2022.
- [10] M. Jeong, M. Park, J. Nam, and B. C. Ko, "Light-weight student LSTM for real-time wildfire smoke detection," *Sensors*, vol. 20, no. 19, p. 5508, 2020.
- [11] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1419–1434, 2018.
- [12] S. P. Hohberg, "Wildfire smoke detection using convolutional neural networks." 2015.
- [13] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2016, pp. 877–882.
- [14] M. Mukhiddinov, A. B. Abdusalomov, and J. Cho, "A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5," *Sensors*, vol. 22, no. 23, p. 9384, 2022.
- [15] B. U. Töreyn, Y. Dedeoğlu, U. Gündükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognit. Lett.*, vol. 27, no. 1, pp. 49–58, 2006.
- [16] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *2004 International Conference on Image Processing*, 2004. *ICIP'04.*, IEEE, 2004, pp. 1707–1710.
- [17] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Stud. Therm. Eng.*, vol. 19, p. 100625, 2020.
- [18] F. Guede-Fernández, L. Martins, R. V. de Almeida, H. Gamboa, and P. Vieira, "A deep learning based object identification system for forest fire detection," *Fire*, vol. 4, no. 4, p. 75, 2021.
- [19] H. A. Hosni Mahmoud, A. H. Alharbi, and N. S. Alghamdi, "Time-Efficient Fire Detection Convolutional Neural Network Coupled with Transfer Learning," *Intell. Autom. Soft Comput.*, vol. 31, no. 3, 2022.
- [20] S. A. Jebur, K. A. Hussein, H. K. Hoomod, & L. Alzubaidi (2023). Novel Deep Feature Fusion Framework for Multi-Scenario Violence Detection. *Computers*, 12(9), 175.
- [21] S. A. Jebur, K. A. Hussein, & H. K. Hoomod (2023). Abnormal Behavior Detection in Video Surveillance Using Inception-v3 Transfer Learning Approaches. *IRAQI JOURNAL OF COMPUTERS, COMMUNICATIONS, CONTROL AND SYSTEMS ENGINEERING*, 23(2), 210-221.
- [22] L. R. Ali, S. A. Jebur, M. M. Jahefer, & B. N. Shaker (2022). Employing Transfer Learning for Diagnosing COVID-19 Disease. *International Journal of Online & Biomedical Engineering*, 18(15).
- [23] L. R. Al-Khazraji, A. R. Abbas, & A. S. Jamil (2023). The Effect of Changing Targeted Layers of the Deep Dream Technique Using VGG-16 Model. *International Journal of Online & Biomedical Engineering*, 19(3).
- [24] <https://www.kaggle.com/datasets/kutaykutlu/forest-fire> .
- [25] S. Alsaidi, "Improving Quality of Video Streaming over Mobile Networks to Leverage Healthcare Services," *IJSER*, vol. 5, p. 62, Mar. 2017.
- [26] R. Ghalib and Z. A. A. Alyasseri, "An optimization method for underwater images enhancement," *Wasit J. Pure Sci.*, vol. 2, no. 2, 2023.