



An efficient fire detection system based on deep neural network for real-time applications

Hitesh Gupta¹ · Neelu Nihalani²

Received: 3 January 2024 / Revised: 11 May 2024 / Accepted: 21 May 2024 / Published online: 1 June 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

Fires represent an important risk to an entire planet, destroying everything from huge cities to impenetrable forests. This can be prevented using fire detection systems, but have been slow to be implemented due to concerns about the high cost, specialized connection, false alarms, and unreliability of existing facility-based detection systems. This take a first step towards utilizing DL to detect fire in images in this work. A Forest Fires dataset, obtained by an UCI ML Repository, is utilized for both training and testing purposes in this study. The components of preprocessing methods involve defining the paths for training and testing data, converting images to pixel representations, normalizing of the data and target variable selection. The model is applied in place due to its ability to highlight intricate details and patterns that are the main elements of precise fire detection. This research presents new methods for detecting forest fires through the use of a carefully selected dataset, transfer learning using the Hybrid (ResNet152V2 and InceptionV3) model and also deep learning based ConvNext model, and innovative preprocessing procedures. The novelty of this study arises from the effective incorporation of Hybrid (ResNet152V2, InceptionV3) model and ConvNext model into the field of fire detection, demonstrating its capability to attain exceptional levels of accuracy and precision. Techniques of visualization and comprehensive evaluation metrics raise the study's novelty. By utilizing the Hybrid (ResNet152V2 and InceptionV3) model, which attains an astounding accuracy, recall, f1-score, and precision of 99.47%, exceptional performance is achieved. While also ConvNext model get 95.53% accuracy. This study makes a valuable contribution to the field of fire detection systems by utilizing innovative deep neural network architectures to enhance performance and dependability.

Keywords Forest fire detection · Deep learning · Neural network · Sensors · Deep fire dataset

1 Introduction

Particularly perilous in today's fast-paced society is arson. As the incidence of fires continues to rise, it is mandatory that all municipal buildings and vehicles be outfitted with

detection and suppression systems. In order to further assure safety of their personnel, many firms conduct fire drills. As a result of this training, would understand what to act on and what to avoid doing in the event of a fire. Forests are widely acknowledged to serve an important role in preserving ecological balance. A forest fire may do a lot of destruction if it starts. FF (Forest fires) are a major threat, yet are sometimes not identified until significant damage has already been done. Possible impossibility of extinguishment. The output shows that it damages the eco-system more than was anticipated. The vast quantities of carbon dioxide (CO₂) released by FF are a major contributor to global warming. Furthermore, it would lead to the final extinction of numerous species that have already vanished [1]. In addition, it might impact the climate, which could trigger disastrous events like floods, earthquakes, and intense rainfall.

A forest is an extensive area characterized by trees, copious amounts of dead leaves, timber, and other similar

✉ Hitesh Gupta
hitesh034@gmail.com

Neelu Nihalani
nihalani.neelu@gmail.com

¹ Department of Computer Science & Engineering, University Institute of Technology, Rajiv Gandhi Proudhyogiki Vishwavidyalaya (Technological University of Madhya Pradesh), Bhopal, MP, India

² Department of Computer Application, University Institute of Technology, Rajiv Gandhi Proudhyogiki Vishwavidyalaya (Technological University of Madhya Pradesh), Bhopal, MP, India

elements. When a fire does start, these materials provide fuel for it. Numerous factors, including extreme heat during the summer months, cigarette smoke, and fireworks at gatherings, all contribute to wildfires. Once a fire has started, it will not go out until it has consumed everything in its path. Early detection of FF can lessen its impact and reduce associated costs [2].

FF is commonplace in a lot of places. These flames seriously threaten human life, the local ecology, and animal inhabitants. Hot, dry weather makes it more likely that a fire will start quickly and spread to nearby vegetation, structures, or anything else. The fire's smoke and heat can be just as lethal. FF is now a significant reason why natural catastrophes have hit so many places worldwide in recent decades. Turkey, Greece, Italy, Algeria, and Morocco have all been hit by the deadliest fires in years, resulting in hundreds of deaths and huge economic damage. By 2021, thousands of fires were already reported, having consumed nearly hundreds of thousands of hectares of land. There have been many casualties and injuries and significant prop-

erty and ecological damage caused by the flames [3]. Many people and things, including lightning, negligent campers, and malicious people, are to blame for FFs. It usually takes many firefighters, police, and forestry professionals to create an FF. Therefore, in this case, FD is crucial. If a fire starts, the address must be determined, and the appropriate authorities must be notified as quickly as possible.

Preventing damage to forests and lives by spotting fires in

time is a top priority. The government employs various FD strategies, including satellite and sensor, tower monitoring, and optical camera usage. In addition to these methods, others may be utilized to extinguish a fire. The most common is utilizing fire to suppress wildfires in dry places or aerial water tanks, as in Canada. These components are swept away and burned in unfueled areas in Middle Eastern nations. But in Australia, just set fire to the area and let it burn out on its own, so there's no risk to humans or animals.

DL might greatly improve the efficiency and precision with which FF is detected. The DL model may be taught to identify signs of FF in the pictures taken above. Drone imagery of the forest canopy has the potential to be timelier and more precise than that captured from the ground. Drones have various advantages over satellite imaging when detecting FFs (the most common and widely utilized aerial imaging method). First, drones can often identify smaller fires than satellites because they may fly lower and gather more precise data. Also, drones are more cost-effective than satellites for many purposes, such as fire detection, agricultural monitoring, infrastructure inspection, and numerous others. And whereas satellites might only be capable of sending out photographs once every few days or weeks, drones are

more flexible and can do so daily. In theory, DL might make it much easier to spot FFs. This article presents an extensive examination of the most recent and refined approach to utilizing DL in order to detect FFs.

The research paper's contribution is as follows:

- The Hybrid (ResNet152V2 and InceptionV3) model and ConvNext model, which are based on DL, is used in this study to classify and recognize images of forest fires.
- The Forest Fires dataset from the UCI ML Repository was used to train and evaluate the suggested model.
- Preprocessing methods are executed, including the definition of training and testing data paths, pixel-to-image conversion, data normalization, and target variable specification.
- Deep performance parameters were attained, including an accuracy of 99.47% with hybrid model and 95.53% accuracy with ConvNext.
- This research makes a meaningful contribution to the field of fire detection systems by implementing a smart deep neural network architecture for reliable failure prediction in area of practical applications.
- The novelty of the research study was achieved by Hybrid network by combining ResNet152V2 and InceptionV3, and ConvNext model in the field of fire detection that enhances the performance and dependability of the system.
- This research can be cited as novel in the branch of forest fire detection, which uses the advanced pre-processing operations and the careful collection of exhaustive data to be utilized. New findings can be observed from combining transfer learning with the chosen (ResNet152V2 and InceptionV3) and ConvNext models because the models improved performance and convergence speed as one of the model's advantages, which is another fact that makes the paper distinct.
- Demonstrating the model's originality to deal with the uniqueness of the forest fire detection problematic, the metrics and visualization methods give a glimpse into the model's performance.
- In the end, the suggested approach shows promise for real-time applications, providing a fresh and innovative way to respond to and lessen the impact of forest fires.

The remainder of the paper is divided into the following sections: Section II provides the literature review of forest fire detection with comparative table, then Section III provide the proposed methodology with their flowchart and algorithm that overcomes the problem of existing work. After this, section IV discusses experimental findings with performance measures and also provides the comparative

analysis between various models. Lastly, in the last section, V provide a paper conclusion & future work.

2 Literature review

Deep Learning was a popular computer-based fire de-tetection method. Many researchers have contributed to designing a fire-detection system. Deep Learning is the field's most notable research. Even though this problem has been argued and studied for years, found few studies and surveys. The following paragraphs summarize our most helpful studies.

In Arteaga, Diaz and Jojoa, (2020), goal to assess a performance of several pre-trained CNN algorithms for classifying FF images on economic development cards such as Raspberry. Increasing occurrences of FF may be attributed to both climate change and irresponsible human behavior. FFs have increased in frequency as a result of elevated global temperatures and a protracted drought brought on by “El Nio” climatic phenomenon. Traditional methods of detecting forest fires, whether from ground or air, are inefficient since they need more time to alert relief troops and careful logistical planning. Recent experiences (the most recent fires) have shown the necessity for increased early detection techniques, leading to conclusion that insufficient action is being taken to address this issue [4].

In Gupta, Liu and Bhanu, (2021), innovative semi-supervised techniques of Spatial and temporal video object segmentation and dense optical flow within a DL framework were presented, incorporating spatially and temporally relevant data. Recognizing this smoke in the cloud without annotated data is, thus, difficult. Dark channel pre-processing is employed in order to diminish the quantity of atmospheric pollution present in video frames, thereby improving the accuracy of detection results. By training on a video before an assessment, they may reduce the time spent on ground gathering truth data. Tests utilizing publicly accessible video datasets demonstrate that suggested approaches outperform past work and are resilient in a wide range of wildfire-prone locales [5].

In Wang et al., (2020), aimed at this situation, An innovative technique is suggested based on DL and dynamic backdrop modeling to reduce false alarms and enable real-time outdoor FFD. SSD (Single Shot MultiBox Detector) DL network traffic was selected for the initial phase of smoke detection. Second, a video's dynamic region was achieved with ViBe dynamic background modeling technology, which considered the smoke's kinetic properties. Third, early findings for smoke detection were improved by using the dynamic region to lessen false alerts. Extensive testing on various forest-themed real-world scenarios showed a

30% accuracy boost compared to a single SSD technique, proving the approach shown here useful [6].

In Jiao et al., (2019), developed a new FF monitoring framework based on CNNs since tiny fire areas are difficult to identify with existing methods. Many sets of FD tests utilizing a self-generated FF dataset and two genuine FF monitor films are undertaken to confirm that the suggested framework may increase the efficacy and accuracy of identifying early FFs. The experimental findings suggest that the framework can efficiently identify the early FF and function in the wide range of demanding fire and illumination circumstances provided in the research [7].

In Priya and Vani, (2019), to improve FD accuracy, a CNN-based Inception-v3-based TL technique is implemented. This technique involves training satellite images, categorizing datasets into fire & non-fire pictures, trying to generate a confusion matrix (CM) to define efficacy of structure, and at last, extracting fire-occurred regions in satellite pictures utilizing local binary pattern to reduce false detection rates [8].

In Kaabi et al., (2018), research developed a method for detecting FFs using YOLOv3 applied to aerial photos collected by uncrewed aerial vehicles. To begin, a UAV platform is built specifically for spotting FFs. Then, making advantage of the onboard hardware's processing power, YOLOv3 is used to construct a small-scale CNN. Around 83% of objects can be identified using this method, and detection may take place at a pace of more than 3.2 frames per second, as shown by our tests. This approach has several benefits when used with UAVs for detecting forest fires in real-time [9].

In Aslan et al., (2019), used an ML-based smoke detector to help stop FF (Deep Belief Network (DBN). Numerous video monitoring and safety systems now incorporate an FD that can identify on camera. To be effective, a smoke detector must have a high detection rate. The method they used to detect smoke was a DBN, essentially an RBM with an extra layer added on top. The smoke-free and smoke-affected areas are concurrently extracted and classified using this method. After measuring the FD rate, pre-training time, and fine-tuning time, they can assess the efficacy of our deployed smoke detection system. The best smoke detection system has the highest detection rate and the quickest pre-training and tuning times [10].

In Pan, Badawi and Cetin, (2020), built a SVM classifier that is trained and evaluated using descriptors extracted from video data that contains smoke and objects colored by smoke [11].

In Gunay et al., (2012), suggested a deep CNN that may be used to detect wildfires using cameras. In order to improve the fire detection rate, they train the neural network

using transfer learning and use a window-based analytic technique [12].

In Elshennawy and Ibrahim, (2020), EADF, an online adaptive decision fusion framework based on entropy functions, has been created for use in computer vision and image analysis [13].

While the existing studies provide a number of methods for detecting forest fires, our suggested model seeks to fill several significant gaps in the literature. To deal with such a wide variety of data types as photos and meteorological data, however, a unified model is required that leverages the best features of deep learning architectures such as Hybrid (ResNet152V2 and InceptionV3) model and efficient pre-processing methods. Some studies also fall short in terms of achieving high precision, recall, and accuracy all at once, or they evaluate model performance using just a subset of available measures. Our suggested model aims to fill these gaps by providing a cohesive and powerful solution, using the strength of ResNet152V2 and InceptionV3 for precise feature extraction and classification.

3 Methodology

The research strategy, problem statement, and sequential procedures that made up the study's methodology are covered in this part. Also included are step-by-step instructions for developing an algorithm and a comprehensive flow diagram of the whole research process.

3.1 Problem statement

Wildfires pose a serious risk to ecosystems and wild-life. Due to rapid climate change, several natural catastrophes of the FF variety have polluted the environment and depleted natural resources [14].

As a result of DL and other ML developments, it is now possible to apply novel methods to the analysis of massive datasets and to previously impossible-to-predict scenarios. DeepFire in the forest is one of our day's most talked-about environmental concerns. The intricacy and unavailability of key elements like humidity, wind speed, etc., mean still have no definitive answers. By utilizing DL, one may side-step these challenges and discover answers with only visual data.

Deep Learning and Machine learning have great promise for resolving several current issues. The limitations of the dataset, in this case, the FF, continue to be the fundamental obstacle to the widespread use of ML to solve practical issues. Researchers that studied data from urban riots, indoor and outdoor fires, and industrial fires encountered the same issue in their efforts to detect wildfires. These

databases lack, however, information about forest ecosystems. Due to this, it appears that FF detection using these datasets may not work well in practice. This study heavily relies on un-balanced and video-based datasets because of their significant success rate; nevertheless, this strategy narrows the scope of the research by providing fewer relevant instances.

3.2 Proposed methodology

This research aims to improve fire detection methods and create a more reliable and timely system. First, may access the Deep Fire datasets directly from UCI Machine Learning Repository. For the categorization issue, I collect and categorize photographs of forest fires and non-fires to help researchers create more reliable techniques for identifying wildfires in the future. The gathered picture collection was then pre-processed by carrying out the necessary procedures. Cropping photos from the Deep Fire dataset in the pre-processing stage might help find the forest areas of interest. When the crop is applied, the photos will be scaled to a standard 250 by 250 pixels. When raw data has been cleaned and sorted, it must be partitioned into test and training sets. The Deep Fire picture dataset was then classified, and the Hybrid model (ResNet152V2 and InceptionV3), also design ConvNext DL approaches were used to train the model and detect fire and non-fire images. After establishing the performance parameters and model testing, evaluate how well the proposed method works. The Fig. 1 shows the proposed flowchart for the fire detection with deep learning models. And their phases discussed here briefly.

Many tactics and procedures are detailed and planned for execution in this part to attain the goals. In addition, this section forecasts the outcomes these operations will produce. The sections that follow outline some of the strategies:

3.2.1 Data collection

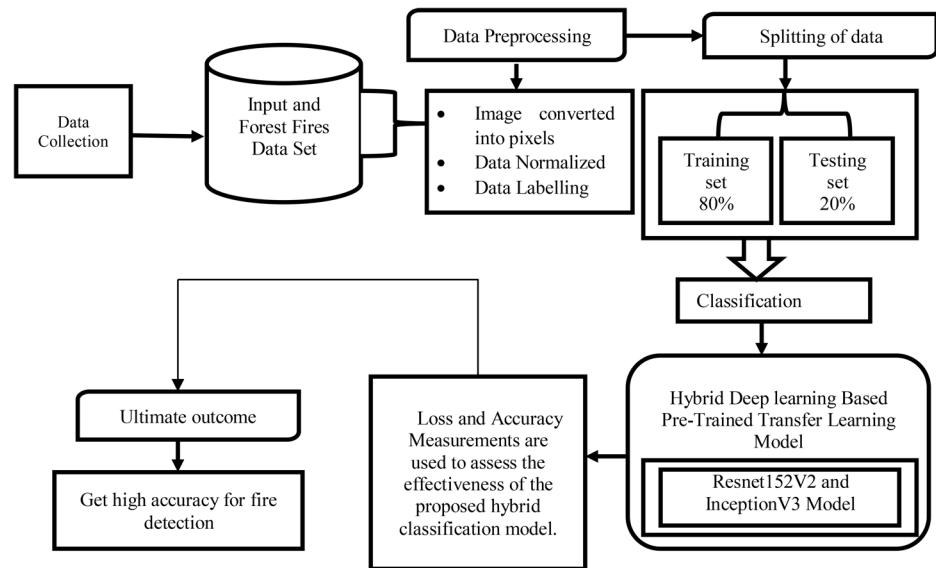
The dataset used in this study for forest fire detection was obtained from the Kaggle data repository.

3.2.2 Data preprocessing

Images uploaded using various search engines and various keywords create the DeepFire dataset¹. Along with the typical forest and mountain scenes, several photos included less desirable aspects like people and fire apparatus. A tidy, well-organized dataset is crucial for effective model training and optimal results. Their DeepFire dataset required extensive preprocessing, including manually identifying regions of

¹ <https://www.kaggle.com/datasets/alik05/forest-fire-dataset>.

Fig. 1 Block diagram of proposed research methodology



interest in each image (forest part). subconsciously got rid of unnecessary details. All photographs were reduced in size to fit into a square of 250 by 250 pixels. The approach quickly and reliably learned the characteristics required for the classification challenge due to these pre-processing steps.

a). **Converting image to pixels.**

A pixel art generator or image pixel converter converts any image into pixel art. Pixelated art refers to a larger image formed by compiling smaller parts (pixelation). The photos were reduced to 250 pixels on each side at this point.

b). **Normalization.**

During image analysis, normalization is performed to adjust the intensity range that each pixel may utilize. Normalization has been accomplished when there is uniformity in data distribution across all input variables (pixels). It allows for quicker convergence during network training.

c). **Defining Target Variables.**

The model's predictions for the target variable are one example. Predictor variables are used to make educated guesses about the occurrence of the target variable.

3.2.3 Data splitting

The choice of an 80% training and 20% testing split in the DeepFire dataset with 1900 photos (950 fire, 950 no-fire) is a standard practice in machine learning. This allocation allows the model to learn from a diverse and representative training set, promoting generalization to new data. The 20%

testing set serves as an independent evaluation, ensuring the model's effectiveness on unseen instances. Maintaining class balance in both sets helps prevent bias and ensures the model's proficiency across both fire and no-fire scenarios.

3.2.4 Classification

In order for DL algorithms to attain high detection accuracy, need a large amount of training data. As a first step towards a more effective forest fire detection system, introduce the ConvNext model and a transfer learning approach based on Hybrid (ResNet152V2 and InceptionV3) model to get better results in forecasting accuracy.

a). **Resnet152V2 model.**

Residual Network (ResNet) [13] is a network structure for CNNs that has many convolutional layers. ResNet outperforms other networks despite its large layer count. The primary distinction between ResNetV2 and its predecessor, ResNetV1, is the incorporation of batch normalization before to each weight layer. A multitude of visual recognition tasks are crucial, as evidenced by ResNet's dependable performance in image identification and localization. ResNet introduces the residual block as a preventive measure against overfitting and to enable the network to advance to the deepest layer. ResNet's layer depths range from 18 to 152, with common values being 34, 50, 101, to 152 [15]. Figure 2 is an example of the ResNet152V2 architecture in action. Architecture of ResNet152V2 is explained in following sub section.

- **Flatten layer.**

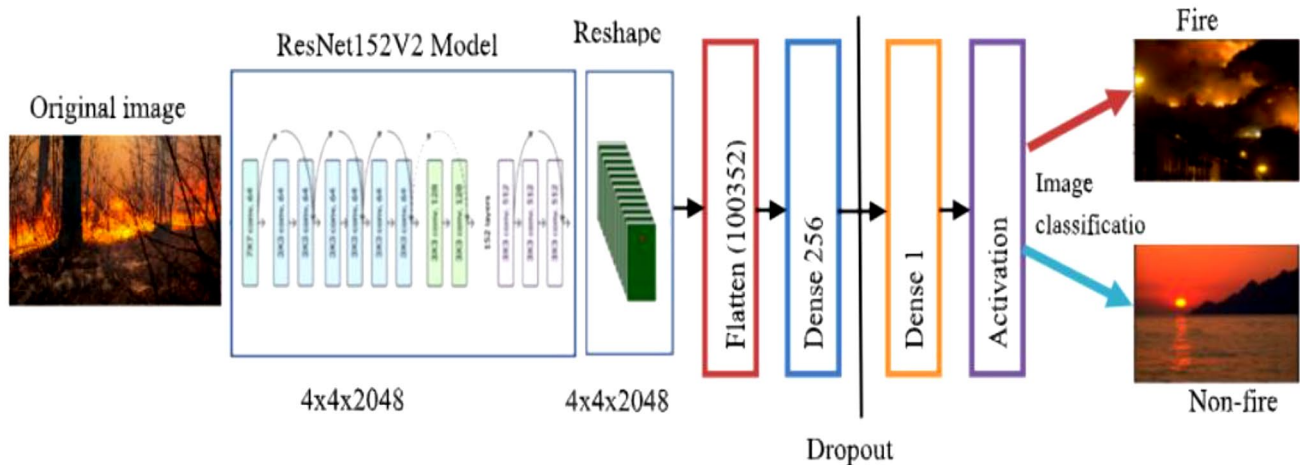


Fig. 2 ResNet152V2 architecture

In a ResNet architecture, the “Flatten” layer typically comes after the convolutional & pooling layers and before the fully connected layers. The number between the brackets, like 100,352, tells you how many neurons or units are in the Flatten layer. This number is based on the size of the feature maps that were made by the convolutional layers that came before it.

- **Dense layer.**

A Dense layer with 256 units can be added to a ResNet152V2 architecture without the need for additional code by placing it after the convolutional and pooling layers but before the output layer. A Dense layer serves as a fully connected layer, capturing higher-level features from the convolutional layers.

- **Dropout Layers.**

ResNet designs do not often make use of dropout layers in the same manner that may reduce in fully connected networks. In highly linked layers, where each neuron is coupled to every neuron in the preceding layer, dropout layers are often used to mitigate the risk of overfitting. Instead, ResNet utilizes skip connections and batch normalization, both of which aid in regularization and help prevent overfitting.

- **Activation function (ReLU and Sigmoid).**

Non-linearity is essential for the optimization process, and activation functions are employed to boost it in a neural network. Our proposed method makes use of ReLU functions and a sigmoid activation function (Equ. 1). The ReLU (Equ. 2) is computationally easy and shows no signs of saturation.

$$R(x) = \max(0, x) \quad (1)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

The supply is x , and the highest of any individual component is 0. That are returned by ReLU(R) after processing the data at its complex characteristics. The logistical function, usually called the sigmoid function, quantifies output probabilities between 0 and 1. the suggested method relied on a detection threshold of 0. If the probability is less than 50%, the answer is 0. Whenever it is more than 50%, the answer is 1.

- b). **InceptionV3.**

InceptionV3 is the new designation for the enhanced version of Inception, formerly referred to as GoogleNet. When comparing its object recognition capabilities, InceptionV3 and its predecessor, InceptionV1, exhibit substantial advancements. The model uses a subset of of the ImageNet dataset in its training phase which ultimately is specialized for the ImageNet LargeScale Visual Recognition Challenge (ILSVRC) [16]. A multi-scale strategy was incorporated in our model. The classifier is the fundamental component, the convolutional block, and the novel Improved Inception module which are the three key components of the InceptionV3 model. With more than 24 million parameters and 48 totals there are for the base learning. Inception-A, Inception-B, and Inception-C are the three main modules of the inception suite, which are the central piece of the whole inception network. The filters size such as 1×1 , 3×3 , 5×5 , 7×7 are shared among the branches of each layer within the module. In order to reduce the number of channels of features and shorten the training process, the 1×1 convolutional kernel is often used [17]. The InceptionV3 architecture is shown in the Fig. 3 below. The architecture has a stacking of Inception-B, Inception-A, and, finally, Inception-C

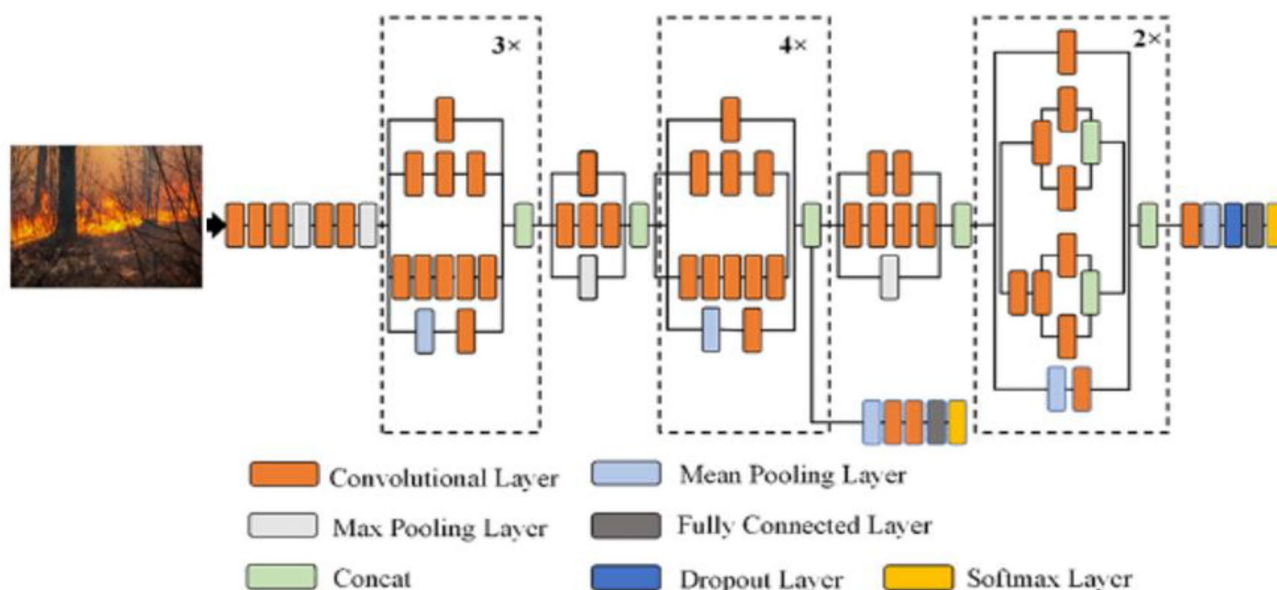


Fig. 3 Architecture of inceptionV3

modules. Channels of the feature map will be equal to 2,048 and its dimensions will be of 8 by 8 after the usage of both convolution and Inception module layers. Then, using the pre-trained model and setting the parameters to match our specific needs will be possible, thanks to three fully connected layers at the end of the Inception modules [18].

c). ConvNext model.

Currently, the image classification algorithm networks are increasing, and efforts are made to further develop deep learning. The Swin transformer, on the other hand, has progressively supplanted CNNs in functions within the coarse-fined classification domain. Subsequently, ConvNeXt, which was enhanced through the implementation of the Swin transformer's inverted bottleneck, depth wise convolution, layer structure, down sampling method, activation function, and data processing method, achieved an even greater degree of classification precision. This network reinstated CNN's significance in image classification [19]. The ConvNeXt [20] architecture is proposed as a temporary substitute for the most advanced transformers currently available. It was created in 2022 by Facebook AI Research (FAIR) researchers. This development's concept is to "modernise" CNNs [21]. ConvNeXt is a convolutional model that is solely based on the Vision Transformers architecture. ConvNet modules are the foundation upon which ConvNeXt is constructed. It is easy to build since it is completely convolutional for learning and testing, while it maintains the efficiency of normal ConvNet. ConvNeXt divides the downsampling layer and has fewer normalisation and activation layers than other backbone networks. The model

was tested using a variety of vision tasks, including object identification and ImageNet classification. In every significant benchmark, it performed better. Convolutions used by ConvNeXt function on a per-channel basis, rearranging just the spatially dimensional information. When the number of clusters in a clustered convolution is equal to the number of input channels, the result is a depth convolution [22].

3.2.5 Finetune process of hybrid (ResNet152V2 and InceptionV3) model

To fine-tune the Hybrid model (combining ResNet152V2 and InceptionV3) for the efficient fire detection system, follow these steps:

- **Load Pre-trained Models:** First, load the pre-trained ResNet152V2 and InceptionV3 models along with their respective weights.
- **Freeze Layers:** Freeze the layers of both models to prevent them from being updated during the initial raining phase.
- **Combine Models:** Create a new hybrid model by combining layers from ResNet152V2 and InceptionV3. This could involve stacking layers from both models or using techniques like model concatenation or averaging predictions.
- **Compile the Model:** Compile the hybrid model using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss function.
- **Fine-tuning:** Fine-tune the hybrid model on the dataset of fire images. This involves training the model for

a specified number of epochs (in this case, 12) with a batch size of 32.

- Evaluate Performance: Using suitable evaluation metrics such as accuracy, precision, recall, and F1-score, assess the performance of the refined hybrid model subsequent to the training process.

3.2.6 Proposed algorithm

Algorithm 1 Forest Fire Detection using Deep Learning.

Input Deep fire dataset D from Kaggle.

Output Classification results indicating fire or non-fire images.

Step:

1. Data Preprocessing.
 - a. Label the dataset.
 - b. Resize images to a standardized size.
 - c. Convert images to a suitable format.
 - d. Normalize image pixel values.
2. Split Dataset.
 - a. Divide the preprocessed dataset into training and testing sets with an 80:20 ratio.
3. Feature Extraction and Classification.
 - a. Initialize pre-trained models: Hybrid (ResNet152V2 and InceptionV3) and ConvNext.
 - b. Set hyperparameters:
 - Number of epochs.
 - Activation functions.

- Optimizer.
- Learning rate.

4. Train models on the training dataset.
5. Evaluation.
6. Evaluate the performance of the trained models using performance metrics:
 - a. Accuracy.
 - b. Loss.
7. Output.
 - a. Obtain classification results.
8. End Algorithm.

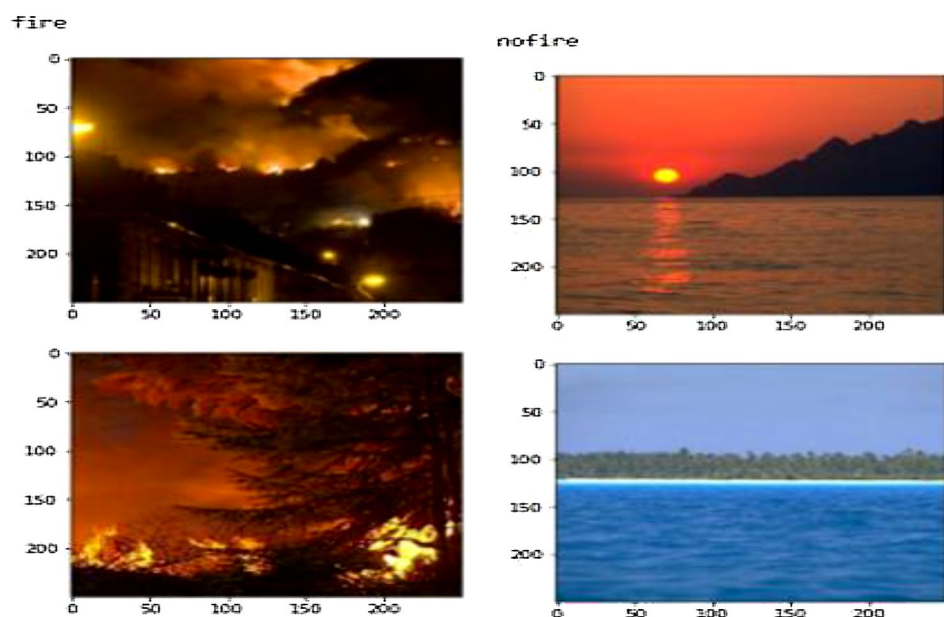
4 Results & discussion

In this part, will go through the dataset, a metrics employed to evaluate performance, and the outcomes of an experiments. This proposed work carried out Python programming tests in a Jupyter notebook.

4.1 Dataset description

For this figure, used a final tally of 1900 pictures from the newly created DeepFire collection, split into two halves. Nine hundred fifty come from the account with fire, and 950 from the account without fire. Figure 4, taken from the DeepFire collection, displays images from both groups.

Fig. 4 Data classes represented by pictures of fire and those without



4.2 Model performance metrics

Measuring performance is essential for the success of machine learning processes. calculate a number that reflects your level of advancement.

4.2.1 Accuracy

The correctness metric allows the algorithm's efficiency to be measured understandably by the user. When a model's characteristics have been established, the accuracy of the model may be computed and expressed as a percentage. The number of correct guesses (expressed by the correct diagonal in the matrix) divided by a total number of samples yields the prediction's accuracy (Calculate as Equ. 3).

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total Equation Number of predictions}} \quad (3)$$

4.2.2 Precision

A label's reliability may be gauged by comparing the observed frequency of positive results with the expected frequency. The precision calculate as Equ. 4.

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (4)$$

4.2.3 Recall

A recall is the proportion of correctly assigned labels to the total number of labels. The recall calculate as Equ. 5.

$$Recall = \frac{\text{true} + \text{ve}}{\text{false} - \text{ve} + \text{true} + \text{ve}} \quad (5)$$

4.2.4 F1 score

It incorporates Recall & Precision metrics into a single value that fully captures their importance. In a perfect world, both Precision and Recall would equal 100%. The f1-score calculate as Equ. 6

$$F - \text{measure} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (6)$$

4.2.5 Confusion matrix

If your machine learning categorization job produces three or more distinct class results, you should use the Confusion Matrix to evaluate your results. It's a spreadsheet with both actual and predicted information. A confusion matrix is a data table that includes both the test data and the real values and is used to evaluate the performance of a classifier.

4.3 Experimental results of hybrid model

The Fig. 5 and table show the proposed models results in terms of plotting graphs, confusion matrix, classification report, and bar graph etc.

A Loss and accuracy graph for the recommended HYBRID (RESNET152V2, INCEPTIONV3) model is shown in Fig. 5. The y-axis displays Loss and accuracy values with time, while the x-axis displays the total period amount. Effectiveness throughout both training and validation is visually shown here. Model get 99.74% validation accuracy and 100% training accuracy, respectively.

Figure 6 displays the recommended HYBRID (RESNET152V2, INCEPTIONV3) model's CM. Comprehend the confusion matrix by considering it a supposition for the null hypothesis. For instance, our data on forest fires demonstrates this categorization in two distinct ways. Genuine and forecasted values are placed against one another in this confusion matrix, with fire and no fire being the two possible states.

- TN= 193; 193 observations from the negative class were deemed safe by the model.
- TP=185, meaning that 185 out of 200 observations (from positive class) were accurately classified as fire by model.
- FN= 1, indicating that model misinterpreted some positive data as negative.
- FP= 1; that is, 2. The model incorrectly labeled some negative data as positive.

Figure 7 shows the testing categorization report for HYBRID (RESNET152V2, INCEPTIONV3) model under consideration. The outcomes for several types of data are shown in this diagram. This chart divides values between two categories: 0 and 1. Precision, accuracy, recall and f1-score at 99% with support 380.

Table 1 below displays the outcomes of the proposed Hybrid (ResNet152V2, InceptionV3) model's test data simulations. There is a 99.47% improvement in f1-score, precision, and recall using the suggested model.

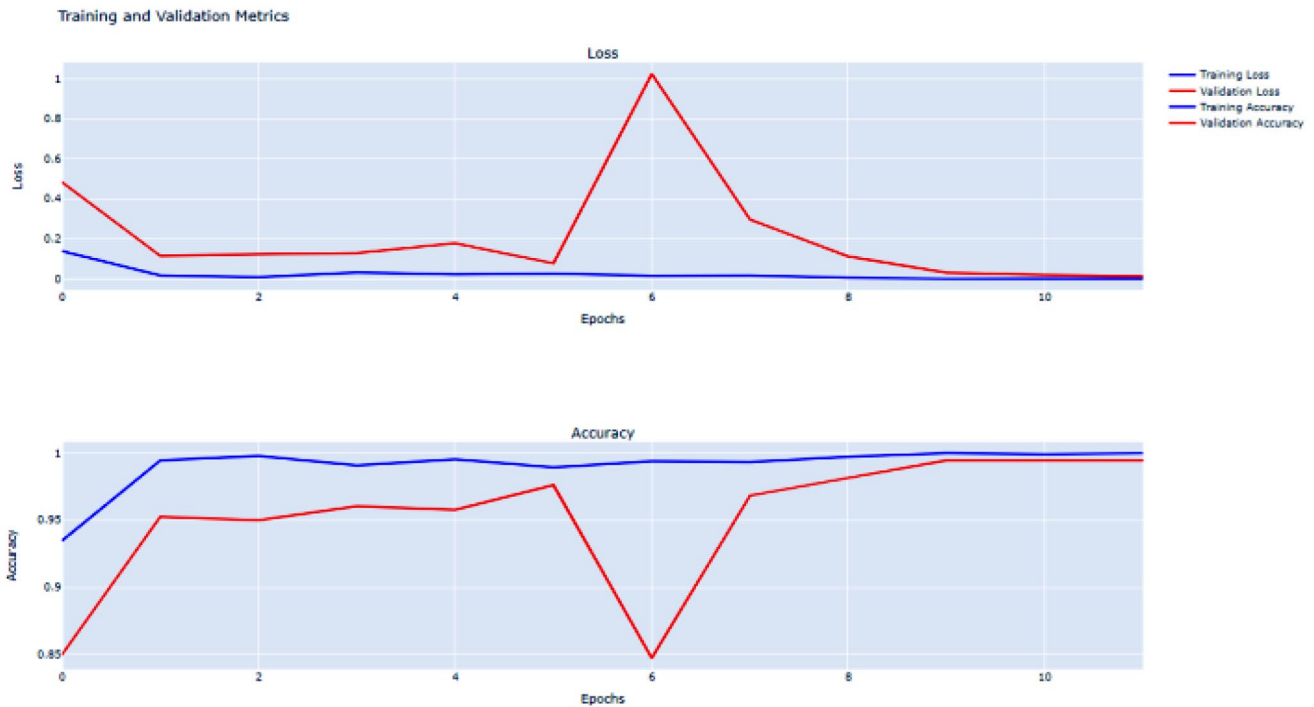
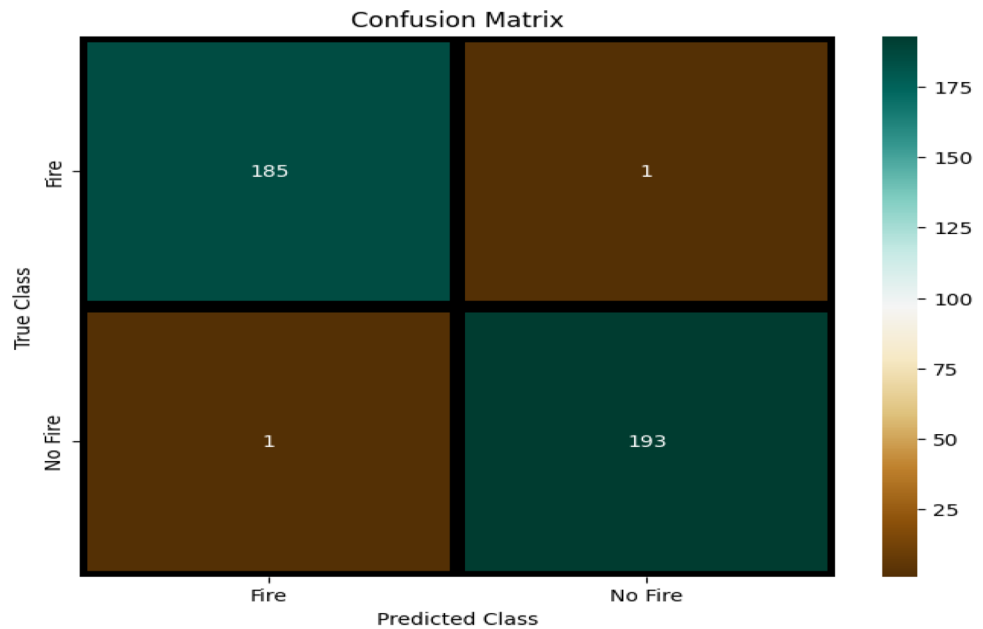


Fig. 5 The loss and accuracy graph for the HYBRID (RESNET152V2, INCEPTIONV3) model

Fig. 6 The HYBRID (RESNET152V2, INCEPTIONV3) model’s confusion matrix under test



4.4 Results of ConvNext model

The following figure and table show the ConvNext model results in terms of plotting graphs, confusion matrix, classification report, and bar graph etc.

The ConvNext model’s accuracy and loss curve for 50 epochs is shown in Fig. 8. In the graphic, the y-axis displays the ConvNext Model’s accuracy and loss values, while the

x-axis indicates the number of epochs. Model get 95.43% and 100% train and validation accuracy.

In Fig. 9 shows the ConvNext Model Confusion Matrix Under Test. Model predicted true positive instance of 185, true negative instance of 178, while false negative instance of 8 and false positive instance of 9, respectively.

In Fig. 10 shows the ConvNext Model Classification Report Under Test. The deep learning based ConvNext

Fig. 7 Classification results of the HYBRID (RESNET152V2, INCEPTIONV3) model tests

	precision	recall	f1-score	support
0	0.99	0.99	0.99	190
1	0.99	0.99	0.99	190
accuracy			0.99	380
macro avg	0.99	0.99	0.99	380
weighted avg	0.99	0.99	0.99	380

Table 1 The test data simulation outcome

Model	ACC	Recall	Precision	F1 Score
Hybrid model	99.47	99.47	99.47	99.47

model 96% classification performance on both classes with support 380, respectively.

Table 2 below displays the outcomes of the ConvNext model test data simulations. There is a 95.53% accuracy, recall, precision and f1-score using the suggested model.

4.5 Comparative analysis and discussion

In this section provides the base (VGG-19, LR, RF, GNB and KNN [14])and proposed (Hybrid (ResNet152V2, InceptionV3) model and ConvNext)models comparison

according to recall, f1-score, accuracy, and precision measure with same dataset. The following Table 3 shows the comparison between base and proposed models for the forest fire detection in terms of performance measures.

The bar graph in Fig. 11 provides a visual comparison of the F1-scores for each model, allowing for a clear assessment of their performance. Each bar represents a specific model, with the height of the bar indicating the corresponding F1-score value. The proposed hybrid model, combining ResNet152V2 and InceptionV3, and the ConvNext model stand out prominently with F1-scores of 99.47% and 95.53% respectively, demonstrating their superior performance compared to traditional machine learning models. Logistic Regression, VGG19, and Random Forest also

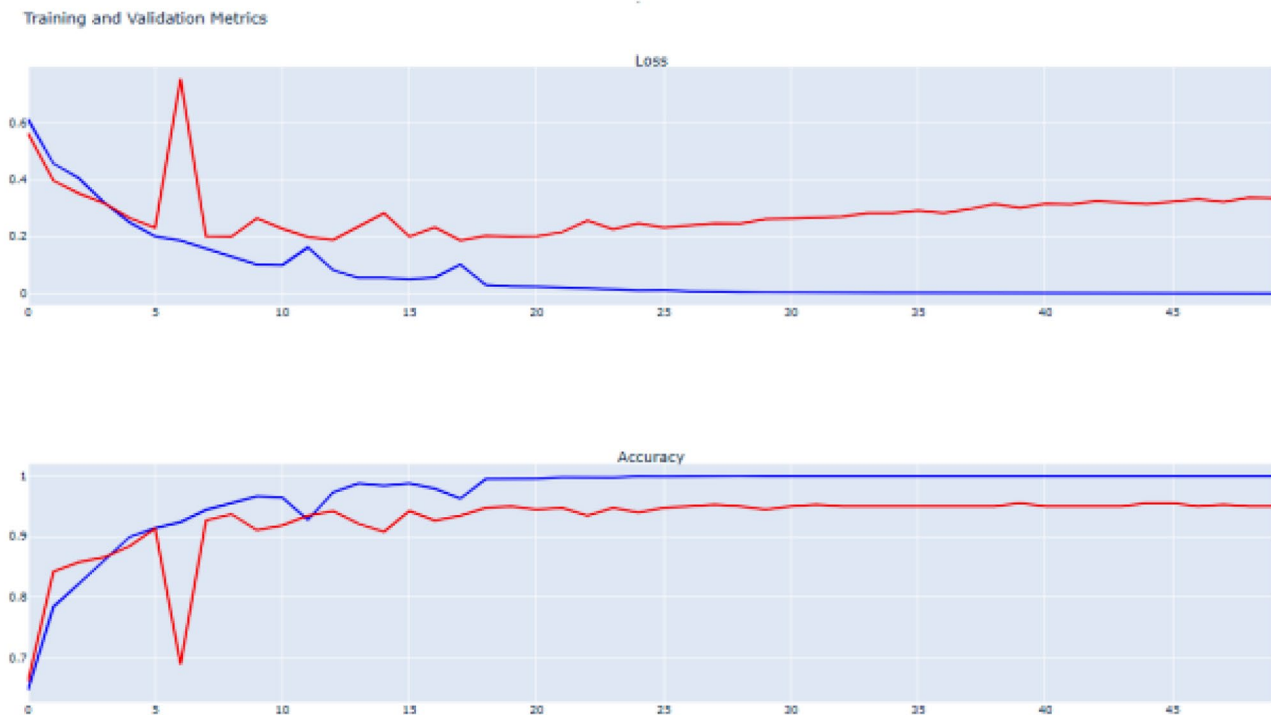


Fig. 8 The loss and accuracy curve of the ConvNext model

Fig. 9 The convNext model confusion matrix under test

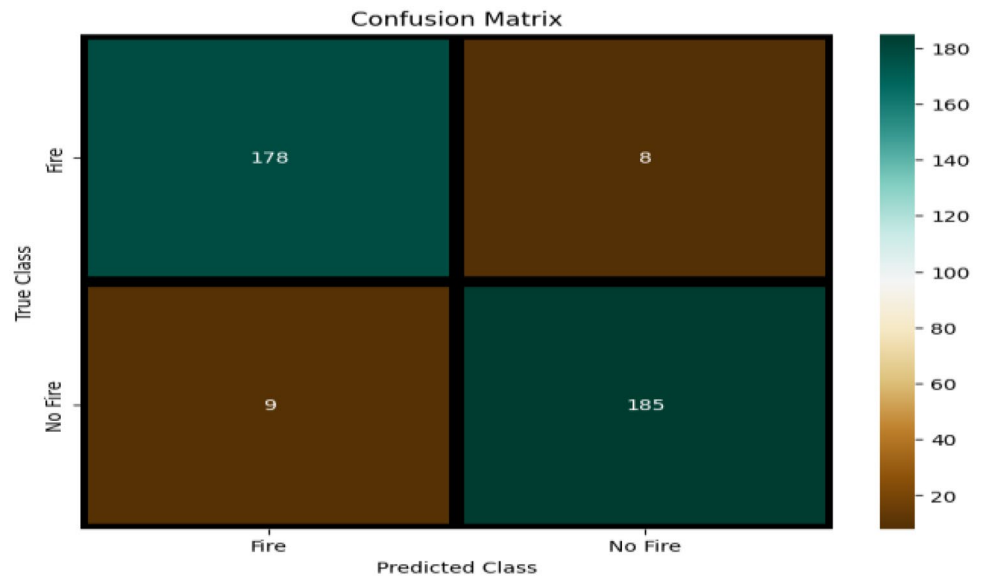


Fig. 10 The ConvNext Model Classification Report of Under Test

	precision	recall	f1-score	support
Fire	0.95	0.96	0.95	186
No Fire	0.96	0.95	0.96	194
accuracy			0.96	380
macro avg	0.96	0.96	0.96	380
weighted avg	0.96	0.96	0.96	380

Table 2 The test data simulation outcome of convNext

Model	ACC	Precision	Recall	F1 Score
ConvNext	95.53	95.53	95.53	95.53

Table 3 Analysis of test data comparisons

Model	ACC	Precision	Recall	F1 score
HYBRID (RESNET152V2, INCEPTIONV3)	99.47	99.47	99.47	99.47
MODEL (proposed)				
ConvNext(proposed)	95.53	95.53	95.53	95.53
LogR	95.06	95	95	95
VGG19	95.53	96	96	96
Random Forest	87.82	88	88	88
SVC	86.5	96	96	96
KNN	86.5	87	87	87
Gaussian NB	81.9	82	82	82

exhibit respectable F1-scores of 95%, 96%, and 88% respectively. However, Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (NB) lag behind with F1-scores of 96%, 87%, and 82% respectively. This graphical representation emphasizes the substantial performance gap between the proposed deep learning models and conventional machine learning approaches,

reinforcing the effectiveness of leveraging advanced neural network architectures for forest fire detection.

5 Conclusion

This study aims, first and foremost, to provide an original approach to FFD and monitoring systems by way of a DL-based FF fighting system. An FFD setup using a Hybrid (ResNet152V2, InceptionV3) model is suggested for this task, where datasets for detecting forest fires were utilized. The information used comes from the publicly available UCI ML repository. For this purpose, must first preprocess the raw data. As a result, you should use the data preparation method. This method specified the flow of training and testing data, pixelized pictures, and standardized measurements, identified the variables of interest, and distinguished between these two data types. Then, the preprocessed data should be divided into training and testing sets. The FFs dataset was used to evaluate Hybrid (ResNet152V2, InceptionV3) model and ConvNext models contrast it to other models. The success of the simulation results is evaluated using a variety of performance assessment metrics, including the confusion matrix, accuracy, recall, precision, and f1-score of 99.47%, were all maximized by suggested

Parameters comparison between base and proposed models

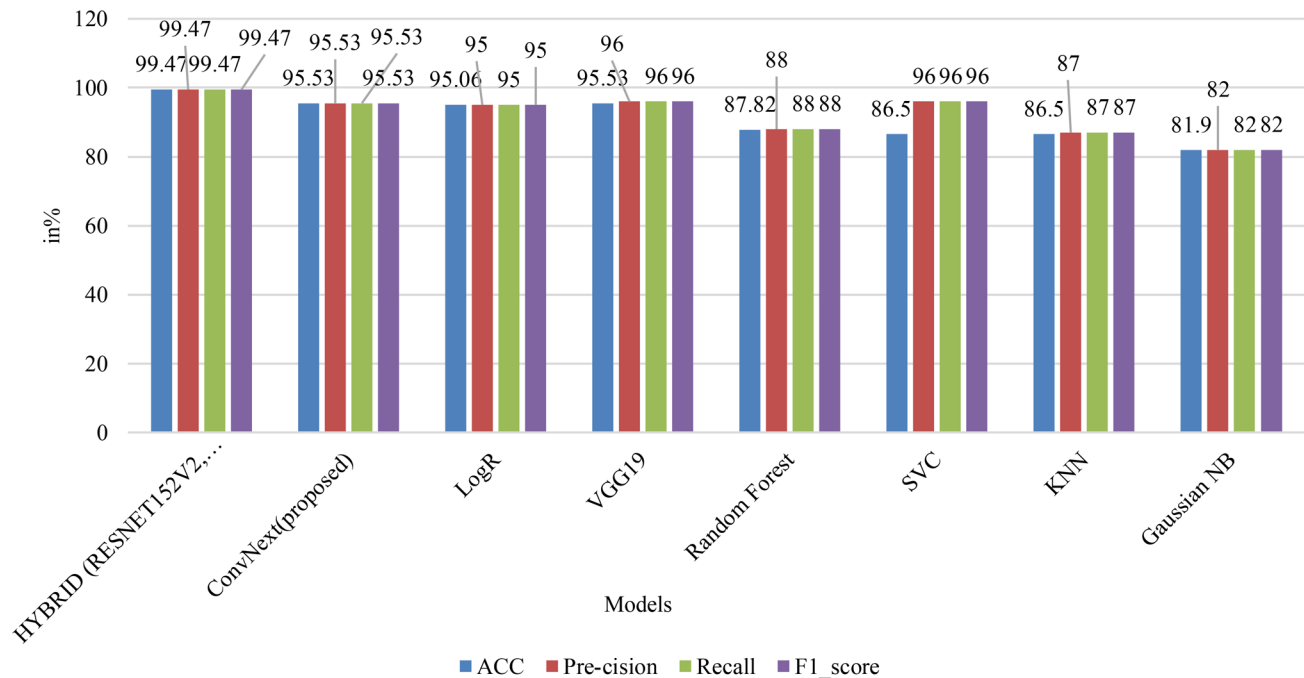


Fig. 11 Bar graph of F1-score comparison between various models

Hybrid (ResNet152V2, InceptionV3) model using validation data. Also, the ConvNext model get 95.53% accuracy for the forest fire detection. The proposed methods outperformed in simulations a number of well-known ML approaches, including GNB (Gaussian Naive Bayes), SVC, NB (naive Bayes), KNN, RF, LR (logistic regression), and VGG19-based transfer learning algorithms (etc.). Fire incidents must be detected quickly and accurately in their early stages, according to recent studies, if are to stop them from spreading. Because of this, plan to expand upon our current research to provide more definitive answers. Our long-term goal is to use the most up-to-date CNN models to reliably and quickly detect fires. They're also interested in learning more about machine learning and multitasking.

Author contributions Both the authors contributed equally to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript.

Funding Not Applicable.

Data availability <https://www.kaggle.com/datasets/alik05/forest-fire-dataset>.

Declarations

Ethical approval Not applicable.

Competing interests The authors declare no competing interests.

References

- Madhwaraj, K.G., Asha, V., Vignesh, A., Shinde, S.A.: Forest Fire Detection using Machine Learning, *IEEE 12th Int. Conf. Commun. Syst. Netw. Technol.*, 2023, (2023). <https://doi.org/10.1109/CSNT57126.2023.10134684>
- Prasanna Rahavendra, A., Praneash, G.P., Rashmika, T., Natarajan, A.: Forest Fire Detection using Computer Vision, in *3rd International Conference on Electronics and Sustainable Communication Systems, ICESC 2022 - Proceedings*, (2022). <https://doi.org/10.1109/ICESC54411.2022.9885250>
- Yandouzi, M., REVIEW ON FOREST FIRES DETECTION AND PREDICTION USING DEEP LEARNING AND DRONES, J. Theoretical Appl. Inform. Technol. (2022)
- Arteaga, B., Diaz, M., Jojoa, M.: Deep Learning Applied to Forest Fire Detection. 2020 IEEE Int. Symp. Signal. Process. Inform. Technol. ISSPIT 2020. (2020). <https://doi.org/10.1109/ISSPIT51521.2020.9408859>
- Gupta, T., Liu, H., Bhanu, B., Early Wildfire Smoke Detection in Videos, in: *25th International Conference on Pattern Recognition (ICPR)*, IEEE, Jan. 2021, pp. 8523–8530. (2020). <https://doi.org/10.1109/ICPR48806.2021.9413231>
- Wang, G., Li, J., Zheng, Y., Long, Q., Gu, W.: Forest smoke detection based on deep learning and background modeling, in *IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, IEEE, Jul. 2020, pp. 112–116. (2020). <https://doi.org/10.1109/ICPICS50287.2020.9202287>
- Jiao, Z., et al.: A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3, in, *1st International Conference on Industrial Artificial Intelligence (IAI)*, IEEE, Jul. 2019, pp. 1–5. (2019). <https://doi.org/10.1109/ICIAI.2019.8850815>
- Priya, R.S., Vani, K.: Deep learning based forest fire classification and detection in satellite images, in *Proceedings of the 11th International Conference on Advanced Computing, ICoAC 2019*. (2019). <https://doi.org/10.1109/ICoAC48765.2019.246817>

9. Kaabi, R., Sayadi, M., Bouchouicha, M., Fnaiech, F., Moreau, E., Ginoux, J.M.: Early smoke detection of forest wildfire video using deep belief network, in *4th International Conference on Advanced Technologies for Signal and Image Processing, ATSSIP 2018*, 2018. (2018). <https://doi.org/10.1109/ATSSIP.2018.8364446>
10. Aslan, S., Gudukbay, U., Toreyin, B.U., Enis Cetin, A.: Early Wildfire Smoke Detection Based on Motion-based Geometric Image Transformation and Deep Convolutional Generative Adversarial Networks, in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (2019). <https://doi.org/10.1109/ICASSP.2019.8683629>
11. Pan, H., Badawi, D., Cetin, A.E.: Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis. *Sens. (Switzerland)*. (2020). <https://doi.org/10.3390/s20102891>
12. Gunay, O., Toreyin, B.U., Kose, K., Cetin, A.E.: Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video. *IEEE Trans. Image Process.* (2012). <https://doi.org/10.1109/TIP.2012.2183141>
13. Elshennawy, N.M., Ibrahim, D.M.: Deep-pneumonia Framework using Deep Learning models based on chest X-Ray images. *Diagnostics*. (2020). <https://doi.org/10.3390/diagnostics10090649>
14. Khan, A., Hassan, B., Khan, S., Ahmed, R., Abuassba, A.: Deep-Fire: A novel dataset and deep transfer learning Benchmark for Forest Fire Detection. *Mob. Inf. Syst.* (2022). <https://doi.org/10.1155/2022/5358359>
15. Thohari, A.N.A., Triyono, L., Hestningsih, I., Suyanto, B., Yobioktobera, A.: Performance evaluation of pre-trained convolutional neural network model for skin disease classification. *JUITA J. Inf.* **10**(1) (2022). <https://doi.org/10.30595/juita.v10i1.12041>
16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2016). <https://doi.org/10.1109/CVPR.2016.308>
17. Ji, Q., Huang, J., He, W., Sun, Y.: Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. *Algorithms*. (2019). <https://doi.org/10.3390/a12030051>
18. Xia, X., Xu, C., Nan, B.: Inception-v3 for flower classification, in: *2nd International Conference on Image, Vision and Computing, ICIVC 2017*, 2017. (2017). <https://doi.org/10.1109/ICIVC.2017.7984661>
19. Li, Z., Gu, T., Li, B., Xu, W., He, X., Hui, X.: ConvNeXt-Based fine-grained image classification and bilinear attention mechanism model. *Appl. Sci.* (2022). <https://doi.org/10.3390/app12189016>
20. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2022). <https://doi.org/10.1109/CVPR52688.2022.01167>
21. Ramos, L., Casas, E., Romero, C., Rivas-Echeverria, F., Moroch-Cayamcela, M.E.: A Study of ConvNeXt Architectures for Enhanced Image Captioning, *IEEE Access*, vol. 12, no. January, pp. 13711–13728, (2024). <https://doi.org/10.1109/ACCESS.2024.3356551>
22. Amjoud, A.B., Amrouch, M.: Object detection using Deep Learning, CNNs and Vision transformers: A review. *IEEE Access*. (2023). <https://doi.org/10.1109/ACCESS.2023.3266093>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.