“FOREST FIRE DETECTION”

Project report submitted by

BHAVANA R [20BSR18008]

Under the guidance of DR.T.RATHA JEYALAKSHMI

Towards the partial fulfillment for the degree of B.Sc. (Data Science and Analytics)

To

Department of Data Science and Analytics School of Sciences

JAIN (Deemed-to-be University) Bangalore

2023

Department of Data Science and Analytics, School of Sciences,
JAIN (Deemed-to-be University), Bangalore.

CERTIFICATE

This is to certify that the present Project titled **"Forest Fire Detection**" has been the outcome of an original study carried out by **BHAVANA R(20BSR18008)** under the supervision of towards the partial fulfilment of the requirements for the degree of B.Sc. Data Science and Analytics of the JAIN(Deemed-to-be University).

This is to further certify that the work reported herein does not form a part of any other thesis/dissertation, on the basis of which a degree, diploma or a certificate has been conferred upon this or any other student in the past.

Dr.T.Ratha Jeyalakshmi	Dr. Arathi Sudarshan
Program Head Department of Data Science and Analytics JAIN (Deemed-to-be University) Bangalore	Head of Department Department of Data Science and Analytics JAIN (Deemed-to-be University) Bangalore

DECLARATION

I, **Bhavana R** hereby declare that this dissertation titled **"FOREST FIRE DETECTION"** has been the outcome of an original study carried out under the guidance of **DR.T.RATHA JEYALAKSHMI** towards the partial fulfilment of the B.Sc. Data Science and Analytics degree of the JAIN (Deemed-to-be University) during the year 2022-2023. This study has not been submitted for any degree, diploma or certificate.

Place: Bangalore

Date:

STUDENT

BHAVANA R
20BSR18008

ACKNOWLEDGEMENT

I take this opportunity to acknowledge the guidance received from our professor, our college administration, and my families and friends towards this exciting journey of researching and working on the final year thesis on "**FOREST FIRE DETECTION**". Completing of this project gives us immense satisfaction, and it would not have been possible without my advisors.

I am indebted to, and sincerely thank our project guide, **Dr. T. Ratha Jeyalakshmi**., professor, department of Data science and analytics, Jain university, for his time, patience, and valuable knowledge, and for leading and guiding me throughout this project.

Additionally, I extend my gratitude to all our teaching and non-teaching staff from the Department of Data science and analytics, Jain university, JC Road campus for their encouragement. We are thankful to **Dr. Arathi Sudarshan**, Head of the Department of Data Science and Analytics for the constant support and encouragement

TABLE OF CONTENTS

Sl. No	Topic	Page No.
01.	Abstract	07
02.	Introduction	08
03.	Literature Survey	09
04.	Existing System	17
05.	Methodology	18
06.	Proposed Frame Work	20
07.	Data Collection	22
08.	Procedure	23
09.	Experimental Results	32
10.	Conclusion	32
11.	Future Scope	33
12.	Reference	34

FIGURE TABLE

Sl. No.	Description	Pg. No
01.	Fully Connected Layer	22
02.	Training the model	26
03.	Loss function and validation loss function plot	28
04.	Image prediction output -01	29
05.	Image prediction output -02	29
06.	Image prediction output -03	30
07.	Image prediction output -04	30
08.	Image prediction output -05	31

ABSTRACT

A forest fire, often known as a wildfire, is an untamed fire that starts in a forest. During the weekends, all the forest fires will wreak havoc on the animal habitat, surrounding environment, and human property. k-means clustering may be used to identify critical hotspots during forest fires to lessen their intensity, manage them, and even anticipate their behaviour. This is useful for assigning the necessary resources. To improve the model's accuracy, it is best to incorporate climatological data to determine the most prevalent wildfire times and seasons.

This research is an effort to detect the presence or commencement of a forest fire in a picture using convolutional neural networks (CNN). The concept is that this model might be used to identify a fire or the beginnings of a fire in a forest using (aerial) surveillance data. The model might be applied in real-time to low-frame rate surveillance footage (assuming flames do not move very rapidly, which is a reasonable assumption) and provide a warning in the event of a fire.

INTRODUCTION

Wildfires are a huge threat to human safety and a significant risk to ecological systems all over the planet. In earlier times, people used to visually check for fire or smoke appearance to detect fire. We employ smoke, a reliable fire indication that is visible before flames, to find fires at an early stage. Considerations should be made about the transparency, responsiveness to environmental conditions, and shape of the smoke. Smoke detection in open spaces is a difficult task that can be made more difficult by the use of sensors, which have temporal and area coverage restrictions. Systems for visual fire detection are utilised to get around this.

To detect fire and smoke, many image processing approaches might be utilised. Images or videos are used as the input for image processing, and the output can either be another image or the parameters or properties of an image. Image processing can be used to carry out a variety of activities, including analysis, classification, feature extraction, and pattern recognition. Several image processing techniques can be used to extract the characteristics and textures of smoke. By employing these methods, hazardous fire-related circumstances can be avoided and public safety can be maintained.

Various technologies are used for forest fire detection, such as satellite imagery, aerial surveillance, and ground-based sensors. These technologies work together to provide a comprehensive view of forested areas and detect any sign of a fire. Once a fire is detected, alerts are sent to the authorities, and fire fighters are dispatched to the location to control the fire.

Due to the tremendous harm they inflict on the environment, property, and human life, forest fires are a reason for concern. Thus, it is essential to find the forest fire early on. This could aid in preserving the area's vegetation, animals, and natural resources. Moreover, it might aid in the early stages of fire control. Due to the wide area and dense forest, monitoring the forests is a challenging undertaking.

The wide-ranging negative effects of forest fires on the environment, the economy, and society, including the deterioration of the forest, includes loss of biodiversity and extermination of flora and fauna , loss of wildlife habitation and exhaustion of wildlife, global warming, loss of valuable wood resources and deterioration of catchment areas.

Forest fire detection has benefited from developments in artificial intelligence and machine learning, with systems using image recognition algorithms to spot smoke or flames in pictures taken by cameras. These technologies can aid in the prompt management of fires by giving early warning of their occurrence.

LITERATURE REVIEW

[1] In this paper, the author uses CNN-convolutional neural networks to detect fire with the help of live video footage through anti-fire surveillance systems. The paper proposes YOLOv2 convolutional neural network is one of the best solutions for detecting fire and smoke both indoor and outdoor environment. You only look once (YOLO) is a deep learning model for object detection, YOLOv2 is the next version which has been upgraded to rectify the setbacks of YOLO namely the inaccuracy to locate and mark the region of interest in the images and the lower recall rate compared to other region-oriented algorithms. Thus, increasing the efficiency of the architecture. They started with an input image of size 128x128x3. They used convolutional layers to map the features on the input image. The features extracted are then given as input to YOLOv2 object detection subnetwork. YOLOv2 Transform layer is implemented to improve network stability for object localization.

[2] This paper proposes that forest fires can be detected by vision-based fire detection systems which can be mounted to an unmanned aerial vehicle (UAVs) for strategically scanning acreage of fire prone areas. This paper also strongly recommends Convolutional neural networks for identifying smoke and fire through videoframes which is taken as images. They have collected the dataset from different internet sources. They have resized the images to canonical size of 240x320. In this paper, the basic idea is to find the fire patches in an image. The authors propose two methods for the algorithm to build the model. First was to apply fire patch classifier from scratch. Second was to teach a full image classifier and apply fine-tuned patch classifier if the image contains fire. Then they compare SVM-pool5 (Support vector machines) with CNN-pool5, the accuracies recorded are 95.6% and 97.3% respectively with a detection rate of 84.8%, making CNN-pool5 network more accurate than SVM-pool5 classifier.

[3]In this paper, Environment can be destroyed by the forest fire, and it could be making a huge amount of loss. Recently, the amazon forest has had a fire and it remained for over 15 days. This resulted a huge loss and it affected negatively to the diversity and global conditions. The wireless sensor networks help in detecting the forest fire.Itcan give a warning as soon as if there any unusual event occurs. Sometimes, these networks can be making false alarms according to the wrong detection. In such cases, machine learning mechanism can be used to prevent such cases. Earlier, satellite-based systems are used to detecting fire. But it may not be possible to finding the distraction as it took pictures of surface of the earth in every two days. As a result, it may not be considered as an effective method. Also, the weather conditions may be affected in the quality of the pictures. Another method for the fire detection was using watch towers. It was handled manually by watching the whole forest area in a tower and finding if there any fire

occurs. Another one is using optical sensors and digital camera. It would not be much effective as the vision can be distracted by the high trees or hills.

[4] Fire can be detected by using the amount of smoke. The smoke sensors are used to measure the amount of smoke from the fire, and it could be compared with a threshold value and if it is beyond that value, it is considered as a fire scenario.Using image processing, fire can be detected as soon as possible. Fixing the CCTV camera everywhere and the images from these cameras can be processed to monitor the fire. If any changes occur, it is easy to detect and extinguish the fire quickly. This system has a water extinguisher for extinguish the fire when the alarm turns on. The CCTV camera is used for recording the video of a particular spot and it is connected to a mini- computer called Raspberry-pi. So that it could get the constant video recording of a particular area. The captured video pictures are processed frame by frame and once the fire detected, the alarm would be turn on. Also, the alarm would be turned off when the fire extinguished completely.The Virtual Network Computing is used for the execution of the program, where the details of video are transferred from the raspberry-pi to the viewing computer.This system includes detection, alert, fire extinguish, software and network modules.

[5] In fire detection, the color of the image from a camera is highly important. Sometimes, it does not possible to watch the entire forest images according to the size as it may be some difficulties in detecting the fire.So that, using Convolutional Neural Network(CNN) technology would be easier to avoid the blindness and accurate level of fire identification. It uses the support vector mechanism for the image classification.In this technique, the image is segmented based on the color of the flame and transferred to the CNN network. This would be found out more attributes and decide there is a fire occurs or not. Fire can be detected by analyzing the color of the flame in a picture. Finding the fire by using the number of pixels plotted in a picture according to the fire color and can be measure the intensity of the fire. So that, it should be easier to detect fire and stamp out the fire. The system should be trained and tested using a large amount of data. Algorithms are used for the segmentation of images and in finding the fire. This method should be more effective and reliable in identifying the fire. The accuracy should be much better than the other methods.(Yuanbin Wang, 2019)

[6] This paper the authors propose a system that mimics the human fire detection system. It uses Faster R-CNN which is a region-based algorithm to detect suspicious Point of interest. After marking the region of interest, the features extracted from the bounding boxes are passed to LSTM Long Short-Term Memory to classify if there is fire or not in short interval of time. Faster R-CNN exploits the features of CNN and introduce a region proposal network which is used to

map the features in the input image. It extracts features through the ROI pooling operation and then classifies according to the class scores of the object position.

[7] In this paper, a novel method for fire detection is proposed based on ensemble learning. The dataset is created using 10581 images from various public sources like BowFire, FD-Dataset, ForestryImages, VisFire. The dataset is preprocessed and fed into not just one but two individual object detectors, YOLOv5 and EfficientDet integrated in parallel mode to achieve better accuracy than a single object detector. Although it uses integrated object detectors, this does not take the whole image into consideration. Therefore, another classifier is introduced to solve this problem. EfficientNet takes the image as whole and evaluates the image to enable total advantage of the information. The results will be decided by a decision strategy algorithm which takes the opinion of the three individual object detectors into account which in turn improves the performance of the model and decrease the rate of False positives. This paper claims that they have achieved a superior trade-off average accuracy, average recall, false positive and latency.

[8] This paper put forward an approach in real-time forest fire detection using wireless sensor network paradigm. This method can detect and forecast the fire more accurately than the other methods used in forest fire detection. Firstly, the sensor networks acquire the details about the humidity, smoke, temperature, and wind speed as these factors affect the forest fire. The sensor nodes are placed widely in the forest, and it is arranged into clusters. The sensor nodes use GPS to track their location as they can sends these location details along with the data such as measurements of temperature to the cluster head. Then, using a neural network method, the cluster header computes the weather index and then these information sends to the manager node. The wind speed is calculated by the wind sensor nodes, which are manually placed in the forest. The users get information from the manager node when an abnormal event occurs like high temperature and smoke. As well as manager node gives information about the levels of forest fire risk rate according to the weather index from different clusters. So that, users can easily find out the exact location of fire in the forest if it occurs. Also, they could protect the forest from the fire hazard due to the early detection (Liyang Yu, 2005).

[9] According to a research method, Light detection and ranging (LIDAR) system is used for the forest fire detection with the help of neural network. LIDAR is mainly used in the environmental and atmospheric studies. A lidar contains a photo detector, radiation emitter, signal receiver and signal processing hardware and software. Here, the neural network is needed to train well with the Neyman-Pearson criterion. The committee machine trained with all possibilities including the false alarm in the validation test sets, to obtain an accurate level of

detection. These committee machines are composed of neural networks. Each committee machine having its' on duty like each one solving significant problems in a recognition problem. Different neural networks can be added together to find solutions to the complex problems as different networks can have different capabilities. In the case of committee machines, two types of neural networks are participated. One is single layer perceptrons, which have many input nodes and a neuron. The other one is using a cascade architecture with two processing neurons where one is connected to the previous neuron and the other one is connected to the input nodes. As a result, the automatic detection of forest fire using committee machine with the help of LIDAR is useful than the traditional ones (Vilar, 2003).

[10] A research study proposes a system which is a combination of using neural networks, computer vision rules, and other expert rules helps in detecting the forest fire. Different approaches are applied to build this system; visual infrared image matching, using the previous hazards memory, image processing, location, size, and geographical data. Here, infrared cameras, visual cameras, meteorological sensors are using for the collection of input data. The image processing tool is combined with the visual and infrared processing. Infrared processing is a combination of detection, oscillation, and alarm processing processes. The growing-region algorithm is used to separate the false alarms. The visual processing finds out the exact location of the visual image from the infrared analysing process. By using different algorithms, it can be detected and easily reject the false alarms. The meteorological information used to detect the humidity, temperature and other factors which affect the forest fire. So that, it is easy to estimate the possibility of fire. Using this proposed system, it can be detecting the forest fire in early stage and avoid the false detection (Begoña C. Arrue, 2000).

[11] Deep learning and wireless sensor network can be helpful in forest fire detection. The research put forward a system using these approaches can detect the forest fire in the early stages. Using the deep learning model, the system detects the fire according to the collection of data from different sensor networks placed widely in the forest. Here, the system consists of the Internet of Things used as a main concept, moving or fixed sensors and a suitable deep learning model. More accurately, there are several sensor nodes places within each 1 km distance and these nodes are transfer data to the internet servers through the gateways. Then this collected information is displayed in a dashboard with online network. Each nodemeasures the values of humidity, carbon monoxide, temperature, carbon dioxide, and atmospheric pressure. These factors have a major role in the forest fire. In this method, firstly, it calculates the weather information from the weather detector located in forest and then find out the Fire weather index (FWI) using the sensor nodes with the help of deep learning algorithms and the metrics. If the FWI have value changes, the Unmanned Aerial Vehicle (UAV) helps to detect these sensor

values more accurately to find the existence of fire. Also, the control tower act as a fire distinguisher to distinguish the fire (Wiame Benzekri1, 2020).

[12] Another research paper presents an idea for the detection of forest fire using spatial data mining and image processing. Firstly, the mining of spatial data occurs and then the digital image from these data is converted to YCbCrColor space and then divided accordingly to identify the areas with fire. A fuzzy set is generated for the fire areas with the values of color space. Color space means a creation, specification, and visualization of colours. The amount of red, blue, and green color determines a color in a computer system. This technology is used in this system. Data mining consists of database, pattern recognition, statistics, machine learning, and visualization techniques. The methods used for the segmentation and identification processes are anisotropic diffusion and the fuzzy logics. Using these rules and approaches, this system detects the forest fire using the spatial data accurately (Prof. K.Angayarkkani, 2009).

[13] In this paper, the authors focus on building a neural network fire alarm systemwith the data collected from the sensor. The sensor measures the temperature, smoke density, CO concentration. The paper proposes a neural network to work on the data obtained from the sensor. The decision-making algorithm use a single detector reading continuously to detect fire or smoke based on a threshold or limit. Radial basis function (RBF) network is used for the object detection. It is type of neural network which generate local response to the input using local approximations. The output is divided into fire, smouldering fire, no fire according to the output of hidden layers of the network.The results of this experimentation shows this system achieved an error rate of 2.3% chance of fire, small fire 1.8%, no fire with 1%. The authors claimthe network can improve it ability to adapt to different unpredictable situations. Further scope of improvements suggested are by collaborating data from different sources.

[14] This research paper, the authors propose a cost-effective fire detection using CNN from surveillance videos. This papers critically analyses the statistics of deaths due to fire. So, their focus is to propose a system that is home friendly and commercial. This paper gives us an insight of how to carefully select the data properly, how to analyse the computational complexity and detection accuracy. They use a model called GoogleNet for extracting the features from the images. For reducing the complexity of larger patches, they reduce dimensionality. The model is tested with two different datasets for validation purposes and results are compared. They achieved an accuracy of 93.5% on the first dataset and an 86% on the next dataset.

[15] P. Piccinini, S. Calderara, and R. Cucchiara proposed a method based on the wavelet model and a color model of the smoke. The proposed method exploits two features: the variation of energy in wavelet model and a color model of the smoke. Smoke is detected based on the decrease of energy ratio in wavelet domain between background and current. The deviation of the current pixel color is measured by the color model. Bayesian classifier is used to combine these two features to detect smoke.

[16] Celik (2007) proposed a generic model for fire and smoke detection without the use of sensors . Fuzzy based approach is used in this system. Color models such as YCbCr, HSV are used for fire and smoke detection. The fire is detected using YCbCr color model samples because it distinguishes luminance and chrominance. Y, Cb, Cr color channels are separated from RGB input image. A pixel is more likely a fire pixel if intensity of Y channel is greater than channel Cb and Cr. HSV color model is used for Smoke detection as is does not show chrominance characteristics as fire. As smoke is the early indicator of fire it should be detected at lower temperature, here its color varies from white-bluish to white, the saturation is low which satisfies the HSV color model property. As like smoke, sky also has grayish color property and it may be identified as smoke. This problem is rectified by Motion Property, where sky will be removed.

[17] Surapong Surit, Watchara Chatwiriya proposed a method to detect fire by smoke detection in video. This approach is based on digital image processing approach with static and dynamic characteristic analysis. The proposed method is composed of following steps, the first is to detect the area of change in the current input frame in comparison with the background image, the second step is to locate regions of interest (ROIs) by connected component algorithm, the area of ROI is calculated by convex hull algorithm and segments the area of change from image, the third step is to calculate static and dynamic characteristics, using this result we decide whether the object detected is the smoke or not. The result shows that this method accurately detects fire smoke.

[18] R.Gonzalez proposed a method to detect fire based on Wavelet Transform. Stationary Wavelet Transform is used to detect Region of Interest. This method involves three steps preprocessing, SWT, histogram analysis. In preprocessing unwanted distortions are removed and image is resized and transformation of resized image is performed. High frequencies of an image are eliminated using SWT and the reconstruction of image is done by inverse SWT. Image indexation is performed to group the intensity colors that are closed to each other. Histogram analysis is used to determine the various levels of indexation. After

analysis a comparison is made with non-smoke frame and non-smoke images are eliminated. These three are combined and fire is detected.

[19] Osman Gunay and Habiboglu [5] proposed a system based on Covariance Descriptors, Color Models, and SVM Classifier. This system uses video data. Spatio-temporal Covariance Matrix (2011) [13] is used in this system which divides the video data into temporal blocks and computes covariance features. The fire is detected using this feature. SVM Classifier is used to filer fire and fire-like regions. This system supports only for clear data not for blur data.

[20] Dimitropoulos (2015) proposed an algorithm where a computer vision approach for fire-flame detection is used to detect fire at an early stage. Initially, background subtraction and color analysis is used to define candidate fire regions in a frame and this approach is a non-parametric model. Following this, the fire behavior is modeled by employing various Spatio-temporal features such as color probability, flickering, spatial and spatiotemporal energy. After flame modeling the dynamic texture analysis is applied in each region using Linear Dynamical Systems, Histogram and Mediods. LDS is used to increase the robustness of the algorithm by analyzing temporal evolution of pixel intensities. Pre-processing is done after this to filter non-candidate regions. Spatio-temporal analysis is done to increase the reliability of the algorithm. The consistency of each candidate fire region is estimated to determine the existence of fire in neighboring blocks from the current and previous video frames. Finally, a two-class SVM classifier is used to classify the fire and no fire regions.

[21] Hamed Adab proposed another system which is based on Indexing. GIS techniques and remote sensing provides further assistance. The indexing may be structural fire index, Fire risk index, Hybrid fire index. Depending on the geographical condition of the area the indexing differs. Validations of indices are based on hot spot data. Structural fire indices show static information and it does not change over short time span and used to predict the risk in advance. Fire risk index changes as the vegetation or climate changes. Hybrid index is a combination of Structure and Fire index. The disadvantage of this indexing is that way of combining.

[22] Akshata & Bhosale proposed another method where Local Binary Pattern acts as a base for fire detection and Wavelet Decomposition is used to detect fire. Pixel level analysis is required in this method. This method uses YCbCr color model to detect fire. Detection is based on threephase. The first phase involves segmentation of image using LBP. LBP is a texture operator whose value is computed using image's center and neighboring pixel values. Further accuracy is improved using Wavelet Transform and complicated data is classified using

this approach. 2D Discrete Wavelet Transform is used for decomposition in this system. 2 images should be used as input and the sub bands of every image are compared with the other, if sub bands are equal the images are same else different.

[23] Cheng (2011) proposed a fire detection system based on Neural Network; here neural network is used in detection information for temperature, CO concentration, and smoke density to determine probability of three representative fire conditions. RBF neuron structure is used, the information regarding temperature, CO concentration, and smoke density are collected and data fusion is used to generate fire signal decision. The detectors have continuous analog outputs, when detection limit is exceeded the hardware circuit sends a local fire indication to fusion center, this force the system detectors to generate final decision. Single-sensor detector is used to generate the final decision.

[24] Zhanqing (2001) proposed another method using NN and Multi-threshold algorithm. In this method the NN not only classify the smoke, sky, background but also generates a continuous random output representing mixture of these. NN consumes time in case of large areas so multi-threshold algorithm also used as well. These two approaches may be combined or used separately depending on the size of the area. Multilayer Perceptron Neural Network is used here. The number of neurons in the output layer is equal to the number of desired parameters of the output vector, which are “smoke,” “sky,” and “background”. The degree of separation between pixels is identified by Euclidean Distance. Multi threshold algorithm is based on channel wise approach, reflectance of each channel value is used for threshold assumption and is applied to each and every pixels of the image, smoke pixels are marked and false pixels are removed. Threshold value is set as 0.9 <= channel 1 reflectance / channel 2 reflectance<= 1.5. Pixels which reach this threshold are smoke pixels else are false pixels and are removed.

[25] Paulo Vinicius Koerich Borges proposed a fire detection method based on probabilistic method and classification. Computer vision based approach is used in this approach. Though this approach is used surveillance it is also used to automatic video classification for retrieval of fire catastrophes in databases of newscast content. There are large variations in fire and background characteristics depending on the video instance. The proposed method observes the frame-to-frame changes of low-level features describing potential fire regions. These features include color, area size, surface coarseness, boundary roughness, and skewness within estimated fire regions. Bayes classifier is used for fire recognition. In addition, apriori knowledge of fire events captured in videos is used to significantly improve the results. The fire region is usually located in the center of each frame. This fact is used to model the probability of occurrence of fire.

EXISTING SYSTEM

Smoke alarms and heat alarms are being used to detect fires. One module is not enough to monitor all of the potential fire prone areas, which is the fundamental drawback of smoke sensor alarms and heat sensor alarms. Being vigilant at all times is the only way to avoid a fire. Even if they are deployed in every nook and cranny, it still won't be enough to constantly produce an efficient output. The price will rise by a multiple as the number of smoke sensors required rises. Within seconds of an accident or fire, the suggested method can generate reliable and extremely accurate alarms. One piece of software powers the entire surveillance network, which lowers costs. Data scientists and machine learning experts are actively conducting research in this area.

There are several existing systems for forest fire detection that use a combination of technologies and techniques to detect and monitor wildfires. They are,

1. Satellite-based fire detection: Satellites equipped with thermal sensors can detect heat signatures and identify the location of wildfires. This technology is useful for detecting fires in remote areas where ground-based detection methods are not feasible.

2. Aerial surveillance: Aircraft and drones equipped with infrared sensors can detect heat signatures and smoke plumes, which can indicate the presence of a fire. Aerial surveillance can cover large areas quickly and provide real-time updates to the authorities.

3. Ground-based sensors: Sensors placed at strategic locations in forested areas can detect changes in temperature, humidity, and smoke density, which can indicate the presence of a fire. These sensors can be linked to a central monitoring system, which can alert authorities in case of a fire outbreak.

4. Image recognition algorithms: Cameras placed in forested areas can capture images and send them to a central system, which uses image recognition algorithms to identify smoke or flames. This system can provide an early warning of a fire outbreak and assist in its timely management.

5. Community-based reporting: In some areas, local communities are trained to report any signs of a fire outbreak. They can use mobile apps or hotlines to report any incidents, which are then verified and responded to by the authorities.

METHODOLOGY

The methodology of forest fire detection involves collecting data from various sources, such as satellites, ground-based sensors, and cameras, and then preprocessing it to remove noise and unwanted signals. Next, features are extracted from the preprocessed data, and a classification model is developed using machine learning algorithms such as decision trees or neural networks to distinguish between normal and abnormal conditions. The model is then used to detect the presence of a fire, and if detected, the system can determine the fire's location and alert the authorities through notifications via email, SMS, or other means. Finally, the system can analyze the data collected during the fire to identify the cause and improve the system's performance in the future. The accuracy and efficiency of the system depend on the quality and quantity of data collected and the performance of the classification model used for detection.

In this paper, the proposed methodology consists of different stages. The stages include A. Acquisition of Dataset, B. Data Preprocessing, C. Feature Extraction, D. Building model, E. Validation and testing.

a. Acquisition of Dataset

Data is in form of video frames which are obtained from footages, but for the ease custom made pictures are to be used to perform training and test. The frames with fire and without fire are then stored as respectively. Then we divide the dataset as training set and test set. This is to be done with great care because if the data fed to the neural network is faulty, the results will be corrupted and fail to produce an accurate system.

b. Data preprocessing

Data preprocessing is the next stage of building a quality machine learning model. Here, the data gets cleaned and processed or simply make the data fit for use. Data preprocessing consist of removing noises and other unwanted objects from the frame. The algorithm must require relevant data otherwise it may produce undesired results.

c. Feature Extraction

For the neural network to accurately detect fire, it needs to know the features of fire, how it looks like in computer's vision. The feature of fire is easily identifiable by human eye. Fire emits reddish color; it has a shape under different circumstances and motion depending on the fuel it uses to burn. In this paper, the shape, color and motion of fire and smoke is used for the detection. We extract the features from different frames in the training set. The neural network extracts these features using feature extraction network in the CNN which is powered by a custom algorithm. After extracting the features these video frames are classified into fire and non-fire scenarios. The features are extracted using bounding boxes using image descriptors.

d. Building the model

The extracted features are then passed to the network to build a model. This model is a set of thresholds to help the network to accurately detect fire. The model learns from the features extracted and set a standard for analyzing new input data.

e. Validation and testing

Validation of the machine learning model is essential because it is clearly important to get the accuracy and see if the system is working. The validation process is executed and according to the test results the system achieved about 97 % accuracy with the validation set.

I. PROPOSED FRAMEWORK

The proposed framework utilizes the advantages of a convolutional neural network. The CNN receives input, it is preprocessed and pools them using region of proposals. Then the region-based object detection algorithm in CNN classifies those proposals into fire and non-fire in the region of interest (ROI) with the help of convolutional layers.

a. Convolutional Neural Networks (CNN)

Convolutional neural networks are a unique class of artificial neural networks that can simulate human brain activity to interpret data via supervised learning. CNN is an appropriately linked network that is a modified multilayer perceptron. It is made up of multiple layers, including an input layer, an output layer, and numerous hidden layers. Convolutional neural networks get their name from their hidden convolutional layers. It provides exceptional object detection capabilities. These convolutional layers evaluate and analyse data using various mathematical models. The subsequent layers receive these outputs from the preceding layers. Given that the network is completely connected, there is a potential of overfitting. In order to prevent this from happening, CNN uses the data's hierarchical structure to sort the data into layers with simpler patterns at the top and more complicated patterns at the bottom. Tensor inputs of various sizes, including height, breadth, and input channels, are provided. The image is now abstract, and using layers, a feature map is created from this abstract image. This is performed several times to mimic the activity of brain neurons. All of the output is filtered and aggregated into a single output in the output layer because the network is fully connected. The size of the feature map directly relates to the number of filters.

b. Architecture

Convolutional layers make to the architecture of a convolutional neural network. Since it can create a region of interest in the original image using image transform filters known as convolutional kernels, CNN differs from other object detection methods. While other techniques use connection weights and weighted sums to form the model. The number of kernels will be the same as the number of feature maps produced. The feature maps' pixel colours correspond to activation sites. Points in the original image with significant activation points are represented by white pixels in the feature map. Black pixels represent strong negative activation points, while grey pixels represent weak activation points. The convolutional kernel converts the reddish orange pixels in the fire zone of the original image to white. A limited portion of the preceding layer serves as the input for each neuron in the convolution neural network. Every neuron in the network produces an output by carrying out operations on the results of earlier levels. The weights of the input values are used to calculate these functions. Convolutional neural networks have the unusual ability to share the same functions across all layers. The feature extractor utilised in the network is known as AlexNet deep CNN, which is a straightforward CNN programme that makes it simple to identify objects in images. The straightforward construction of convolutional neural networks is shown in Fig.

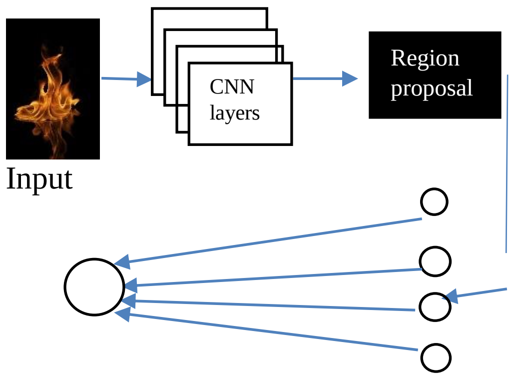

Fig - 01 Output fully connected layers

Architecture of CNN

The data is provided as input, in this case, photographs of fire, and the aforementioned diagram, depicts the fundamental architecture of convolutional neural networks. The image is then abstracted by the network's layers, which also remove any unnecessary background noise and highlight the object that needs to be detected. The output from the layers is analysed by the decision-making algorithm to arrive at a result. The layers generate a region of proposals that are then integrated to create a machine learning model in the fully linked layers.

II. DATA COLLECTION

The network is trained on a dataset that comprises photos classified as 'fire' or 'no fire'. Most of these photographs depict woodland or forest-like landscapes. Photographs tagged 'fire' had visible flames, whilst images labelled 'no fire' were taken in woods. The data augmentation methods offered by Keras are used to apply several random transformations (zooms, shifts, cropping, and rotations) on photos before they are supplied to the network in order to train a network that generalizes well to new images.

The Forest Fire Detection dataset is a valuable resource for researchers and practitioners working on the development of machine learning algorithms for detecting forest fires. The dataset can be used to train and evaluate machine learning models for the task of fire detection, which is critical for preventing and minimizing the impact of forest fires.

The dataset can also be used to study the impact of different factors, such as weather and lighting conditions, on the performance of machine learning algorithms for detecting forest fires. This can help in the development of more robust and accurate fire detection systems.

III. PROCEDURE

1. Importing library

   By importing the required libraries, the code allows TensorFlow and Keras to be used to create a deep learning model for picture classification.
   For mathematical calculations and tensor manipulation, tensorflow and numpy are loaded. For operating system-related functions, os is used. The OpenCV library used for image processing is called cv2. For image preparation and augmentation, ImageDataGenerator and an image from tensorflow.keras.preprocessing are utilised. matplotlib.The performance of the model and the data are visualised using pyplot.

   The deep learning model's architecture is then put up in the code, which is not displayed in this code sample. Given that convolutional neural networks (CNNs) are frequently employed for image classification applications, it is likely that the model is a CNN.

   Data augmentation, that increases the amount of data available for training the model and helps avoid overfitting, is done using the ImageDataGenerator. Individual photos are loaded and prepped for inference using the image module. Finally, the performance of the model and the photos are visualised using matplotlib.pyplot. This could involve visualising the activations of the CNN's layer activations or showing the training and validation loss and accuracy with time.

2. Converting the pixels in range [0,255] to range [0,1]
   Before feeding the input data into machine learning and deep learning models, normalising the data is a frequent practise. Normalisation is primarily done to speed up and improve the stability of the model's optimisation process.
   ImageDataGenerator objects utilised for data preparation and augmentation are the train and test variables. The rescale parameter is used to scale pixels in images with values between [0, 255] and [0, 1]. This is accomplished by dividing each pixel's value by 255, the highest value a pixel in an 8-bit image may have.
   We make sure the model is trained on input data with a consistent scale by rescaling the pixel values to be between [0, 1]. As a result, training becomes more efficient, and the model's convergence improves. Furthermore, it stops the gradients from growing too big during backpropagation, which might result in unstable optimisation.

3. Pre-processing and Training and Test set creation

In this code creates data generators that import and prepare photos from directories in a format appropriate for deep learning model training. Image Data Generator objects that are utilised for data preparation and augmentation are the train and test variables. To generate data generators for training and testing, use the train.flow_from_directory() and test.flow_from_directory() methods, respectively. These techniques produce batches of augmented images on the spot while training, using the path to the directory holding the images as input.The photos are resized to a fixed size of 150 × 150 pixels using the target_size argument. This guarantees that every image is the same size, which is necessary for deep learning model training.

The batch_size parameter controls how many photos are included in each data batch. To balance the trade-off between model accuracy and processing resources, this value can be changed. The batch_size parameter controls how many photos are included in each data batch. To balance the trade-off between model accuracy and processing resources, this value can be changed.
As this is a binary classification problem, where each image is either a forest fire image or a non-forest fire image, the class_mode option is set to "binary." 'categorical' for multi-class classification issues or'sparse' for integer-encoded class labels are some more class_mode variables that are feasible.

This code creates data generators that load and prepare the training and test images in a way that is appropriate for deep learning model training. In order to avoid overfitting and enhance the model's capacity to generalise to new, unknown photos, the data generators developed will be utilised to supply the model with enhanced images during training and testing.

4. Testing Validation

To obtain the dataset's mapping of class names to associated index values, use the test_dataset.class_indices step. In order to label the classes in the dataset, it returns a Python dictionary that maps the class names to their corresponding integer values.

In the context of this code, the flow_from_directory() method was used to import and preprocess test photos and generate a data generator object called test_dataset. This object's class_indices attribute returns a dictionary with the class names (in this case, "fire" and "non-fire") as keys and the integer labels (zero for "non-fire" and one for "fire") as values.

It's crucial to assess how well the trained deep learning model performs using this mapping of class names to integer labels. The predicted class is chosen by choosing the class with the highest probability score once the model has been trained, which generates a probability score for each class. We may analyse and assess the model's predictions by converting the predicted integer label back to its appropriate class name using the class indices, the test_dataset.class_indices step provides a dictionary that maps the class names to the test dataset's corresponding integer labels. This dictionary is crucial for assessing how well the trained model performs on the test dataset's unknown classes.

5. Building CNN Model

This step defines a deep learning model using the Keras Sequential API for binary classification of forest fire images. The model architecture consists of multiple convolutional and pooling layers, followed by a fully connected (Dense) layer and an output layer.
The first layer is a Conv2D layer with 32 filters, a kernel size of (3,3), and a Rectified Linear Unit (ReLU) activation function. This layer expects input images with a shape of (150,150,3), where 3 represents the RGB channels. The input_shape parameter specifies the input shape of the first layer.
The next layer is a MaxPool2D layer with a pool size of (2,2). This layer reduces the spatial dimensions of the output from the previous layer by a factor of 2.
The next three layers (Conv2D, MaxPool2D, Conv2D) follow the same pattern as the first two layers, but with more filters and a larger kernel size. The number of filters increases from 32 to 64 to 128, while the kernel size remains at (3,3). These layers continue to learn higher-level features from the input images and reduce the spatial dimensions of the output.
The final Conv2D layer has 128 filters and a kernel size of (3,3), and is followed by a MaxPool2D layer with a pool size of (2,2). This layer extracts the most abstract features from the input images and further reduces the spatial dimensions of the output.
The next layer is a Flatten layer, which converts the output of the previous layer to a 1D vector. This layer is necessary to connect the convolutional layers to the fully connected layers.
The next layer is a Dense layer with 512 units and a ReLU activation function. This layer learns the high-level representations of the features extracted by the convolutional layers.
Finally, the output layer is a Dense layer with a single unit and a sigmoid activation function. This layer produces a probability score for the input image being a forest fire or not. The sigmoid function is used to squash the output score between 0 and 1, which can be interpreted as the probability of the input image belonging to the positive class (forest fire).

6. Compiling the Model

This code compiles the deep learning model that was defined previously using the Keras compile() function. The compile() function is used to specify the optimizer, loss function, and evaluation metrics, which are necessary for training the model.

The optimizer parameter selects an optimization algorithm to update the model's weights during training. The Adam optimizer is used in this code, which is a popular algorithm that adjusts the learning rate based on the gradients of the model parameters.

After compiling the model, it can be trained on the training dataset using the fit() function. During training, the model is iteratively adjusted to minimize the loss function by updating the weights and biases of the neurons in the network based on the gradients computed during backpropagation. The evaluation metric is calculated at the end of each epoch to monitor the performance of the model on the training dataset.

7. Fit the Model

This step utilises the fit() function of the an object to train a machine learning model. The model was trained for 10 epochs (iterations over the training data) using the train_dataset, which contains the training data. Throughout the training process, the model is tested against new data to gauge its success. To visualise the model's accuracy and loss on the training and validation data, use the code line that returns the history of the training process (r).

```
In [9]: r = a.fit(train_dataset, epochs = 10, validation_data = test_dataset)

Epoch 1/10
58/58 [==============================] - 80s 1s/step - loss: 0.2817 - accuracy: 0.8843 - val_loss: 0.2842 - val_accuracy: 0.911
8
Epoch 2/10
58/58 [==============================] - 77s 1s/step - loss: 0.1357 - accuracy: 0.9563 - val_loss: 0.3225 - val_accuracy: 0.867
6
Epoch 3/10
58/58 [==============================] - 78s 1s/step - loss: 0.1538 - accuracy: 0.9438 - val_loss: 0.2009 - val_accuracy: 0.941
2
Epoch 4/10
58/58 [==============================] - 74s 1s/step - loss: 0.1062 - accuracy: 0.9640 - val_loss: 0.3560 - val_accuracy: 0.867
6
Epoch 5/10
58/58 [==============================] - 77s 1s/step - loss: 0.0995 - accuracy: 0.9672 - val_loss: 0.1556 - val_accuracy: 0.941
2
Epoch 6/10
58/58 [==============================] - 73s 1s/step - loss: 0.1046 - accuracy: 0.9716 - val_loss: 0.1181 - val_accuracy: 0.941
2
Epoch 7/10
58/58 [==============================] - 75s 1s/step - loss: 0.0605 - accuracy: 0.9782 - val_loss: 0.0969 - val_accuracy: 0.955
9
Epoch 8/10
58/58 [==============================] - 71s 1s/step - loss: 0.0602 - accuracy: 0.9787 - val_loss: 0.2839 - val_accuracy: 0.926
5
Epoch 9/10
58/58 [==============================] - 73s 1s/step - loss: 0.0555 - accuracy: 0.9803 - val_loss: 0.0753 - val_accuracy: 0.970
6
Epoch 10/10
58/58 [==============================] - 72s 1s/step - loss: 0.0518 - accuracy: 0.9825 - val_loss: 0.0714 - val_accuracy: 0.970
6
```

Fig 02. Training the model

8. Predictions to work on Testing Dataset

This step is making predictions using a model 'a' on a test dataset. The predicted values are initially stored in the 'predictions' variable. However, these predictions are in decimal form, so in the next line of code, the 'np.round' function is used to convert them to integers by rounding off to the nearest whole number. This is often done when predicting binary outcomes, where the predicted values are usually between 0 and 1, but we want a result that is either 0 or 1 (e.g., for classification problems).

9. Printing the length of predictions

The number of elements in the list of predictions is provided by this step. A Python built-in method called len() returns the number of elements in an object. The item in this scenario is a list of predictions, and len(predictions) yields the list's element count. This reveals the number of predictions that were made.

10. Plotting the loss

Plotting the loss refers to creating a graph of the loss function (also known as the cost function) during the training process of a machine learning model. The loss function measures the difference between the predicted output of the model and the actual output. The goal of training a machine learning model is to minimize this loss function, which is achieved by adjusting the model's parameters during training.

By plotting the loss function over time, we can visualize how the model is learning and improving over the course of training. Ideally, the loss function should decrease as the model improves its prediction accuracy. However, if the loss function increases or reaches a plateau, it indicates that the model is not learning effectively and further adjustments may be necessary.

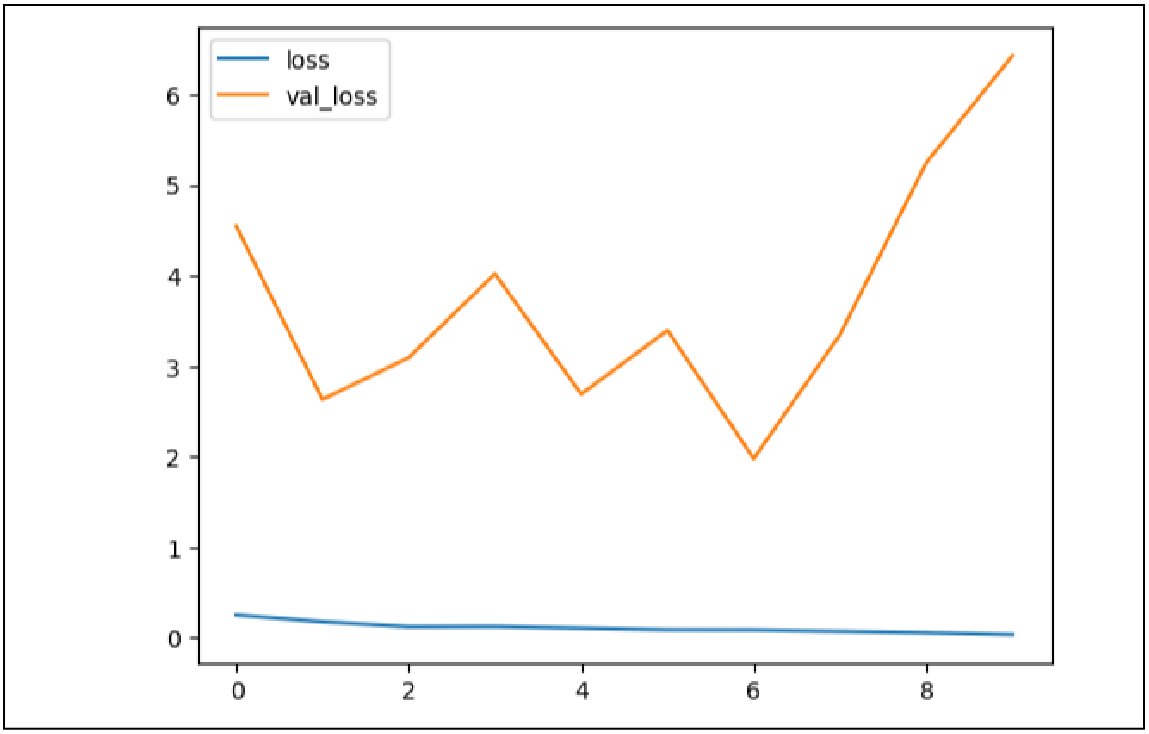

Fig. 03 Loss function and Validation loss function plot

11. <u>Taking the individual images from the dataset, loading and checking the result</u>

This step defines a function called predict Image that takes a filename as input. The function loads an image from the specified file, resizes it to 150x150 pixels, and converts it to a numpy array. The function then makes a prediction on the image using a pre-trained model (stored in the variable "a") and prints the prediction. If the prediction is 1, the function displays the label "No Fire" on the image plot. If the prediction is 0, the function displays the label "Fire" on the image plot. This function is used to predict whether an image contains a fire or not based on a pre-trained model.

12. <u>Predicting image whether it as Fire or not</u>

Using image classification algorithms, we determine whether an image contains fire or not. A dataset of photos with labels indicating whether or not they include fire can be used to train a machine learning model. The model gains the ability to recognise visual cues in the images that separate the two classes.

Once trained, the model may be used to forecast whether or not new photos will include fire. To uniformize the image's size and format, preprocessing is done first. The image is next analysed by the model, which gives each class—fire or non-fire—a likelihood score. The predicted categorization of the image is the class with the highest likelihood score.

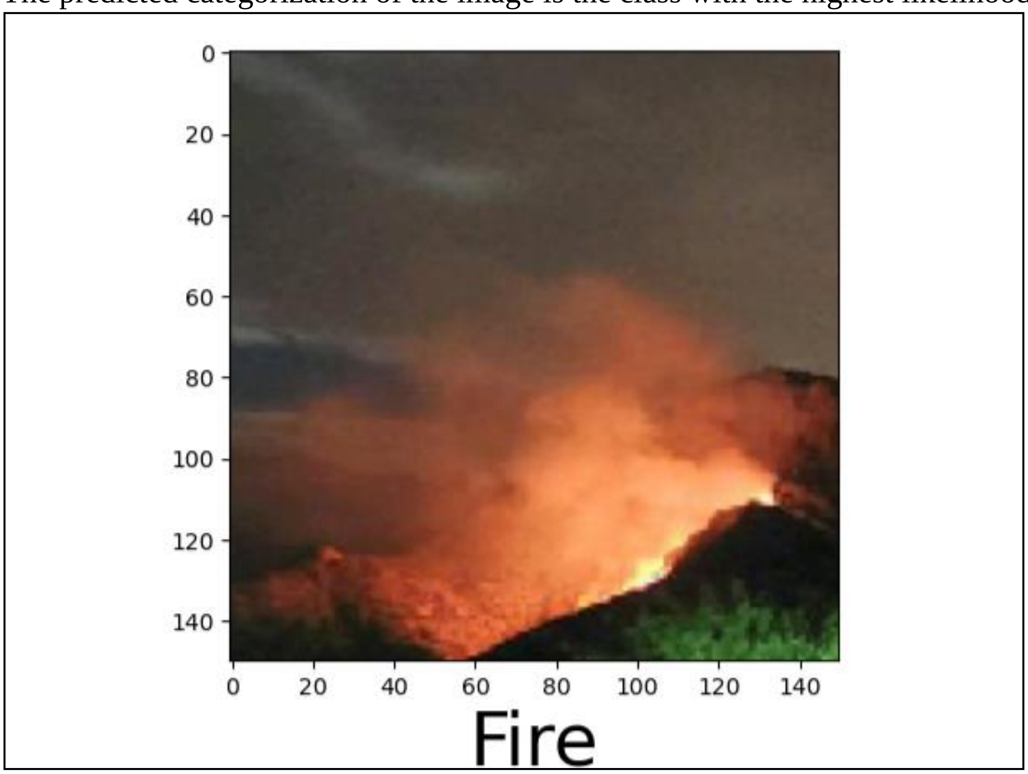

Fig. 04 Image prediction output -01

Fig. 03 displays the output with the word "fire" in it. We can observe the spread of fire in the photograph, and our model correctly predicted it and indicated the presence of fire.

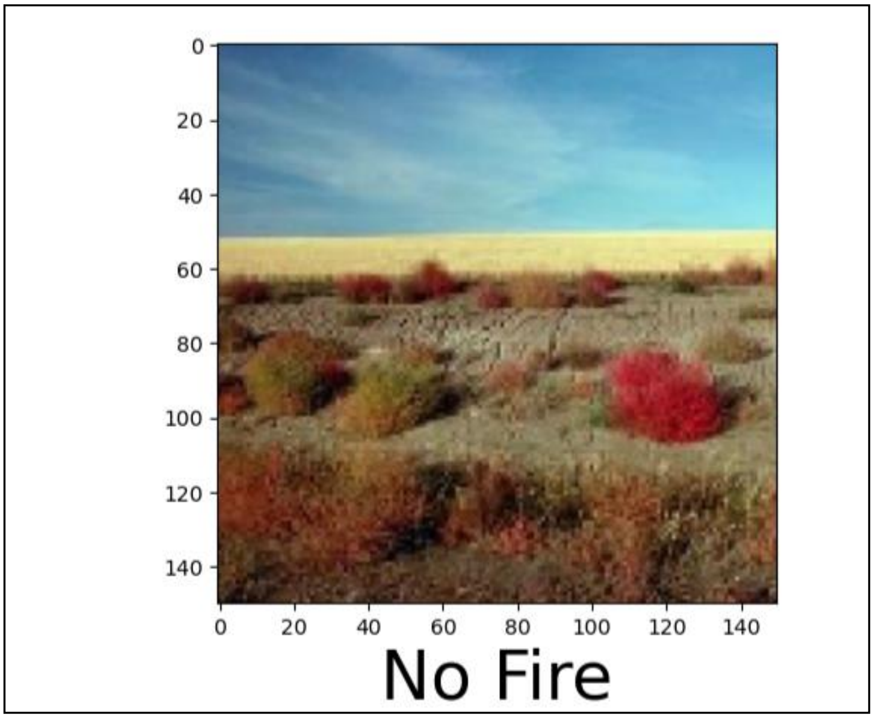

Fig. 05 Image prediction output -02

The output with the word "no fire" in it is shown in Fig. 04. Subsequently our model correctly predicted that there would be no fire, we can see that there isn't any in the image. Even there is presence of colour red(red flowers).

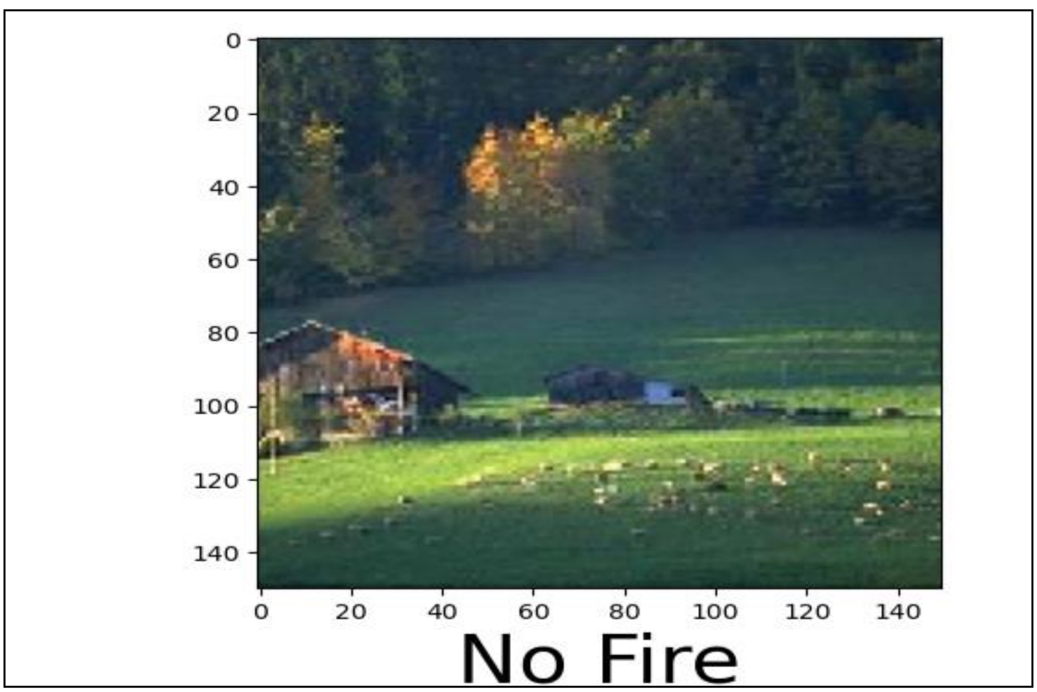

Fig. 06 Image prediction output -03

The output that contains the phrase "no fire" is displayed in Fig.05. Therefore, as we can see in the image, our model properly predicted that there was unlikely to be any fire.

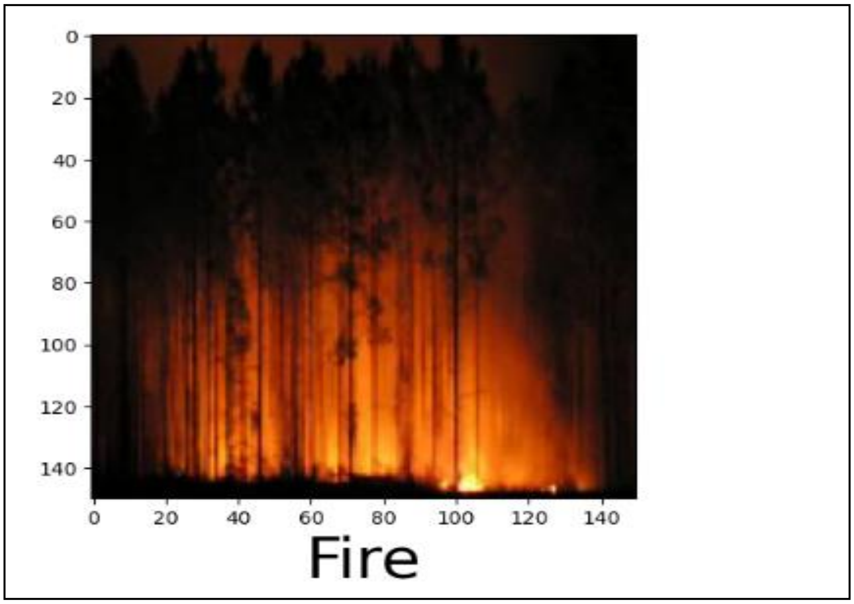

Fig. 07 Image prediction output -04

The output with the word "fire" in it is shown in Fig. 6. In the image, we can see how the fire is spreading, and our model properly anticipated this and indicated that there was fire in the image

```
predictImage("C:/forest_fire/Testing/images/abc356.jpg")

1/1 [==============================] - 0s 50ms/step
[[1.]]
```

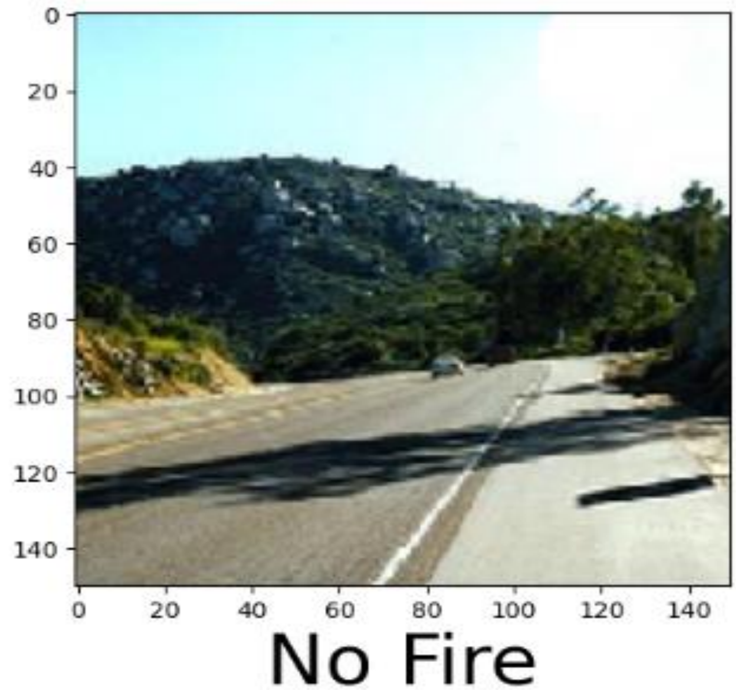

Fig. 08 Image prediction output -05

In the image shown in Figure 7, we can see a road, but there is no sign of a fire, and our model likewise indicates that there isn't one.

IV. EXPERIMENTATION RESULTS.

The findings of the project are greatly satisfying. The system detected fire with an accuracy rate of 97 %. The result obtained show promise for implementation of Convolutional neural networks for detecting fire compared to other neural networks. The system combines several training data intelligently for calculating and reduce false alarm rates with [fully connected network. Then this data is passed to decision-making algorithm to classify whether there is a fire or not. Although it has minor detection errors in some images, the overall performance and statistics are super-efficient.

CONCLUSION

The scope the detection of forest fire is challenging as well as innovative. If this system with less error rate can be implemented at a large scale like in big factories, houses, forests, it is possible to prevent damage and loss due to random fire accidents by making use of the Surveillance systems. The proposed system can be developed to more advanced system by integrating wireless sensors with CCTV for added protection and precision. The algorithm shows great promise in adapting to various environment.

FUTURE SCOPE

To increase accuracy and response times, forest fire detection systems are projected to include cutting-edge technology like machine learning, artificial intelligence, and deep learning algorithms. Early detection and monitoring of forest fires in remote locations can be aided by integration with satellite and drone imagery.

Real-time monitoring of areas that are prone to fire and the early discovery of forest fires are made possible by the deployment of IoT devices like sensors, cameras, and weather stations. To predict the occurrence of forest fires, predictive models based on historical data and weather patterns can be created. In addition to detection, a bigger focus will be placed on prevention using strategies including controlled burning, defensible space, and fire-resistant building materials.

Citizens can report and monitor forest fires using mobile apps, which can speed up reaction times and coordinate efforts with authorities. Governments, non-governmental organisations, and communities will work together more to prevent and control forest fires.

Drones and other autonomous systems can be used to monitor and put out forest fires, lowering the hazards to human fire fighters. Including emergency response systems: Systems for responding to emergencies, such as fire fighting crews, medical assistance, and evacuation protocols, can be connected with systems for detecting forest fires. The use of fire-retardant gels and foams, as well as the deployment of aircraft and ground-based fire fighting equipment, are examples of advanced fire fighting techniques that can help manage forest fires more effectively.

REFERENCE

[1] LEARNING TO PREDICT FOREST FIRES WITH DIFFERENT DATA MINING TECHNIQUESment. . .: LEARNING TO PREDICT FOREST FIRES WITH DIFFERENT DATA MINING TECHNIQUES. (n.d.). Retrieved from https://www.researchgate.net/publication/228527438_Learning_to_predict_forest_fires_with_different_data_mining_techniques

[2] Lammey, B. (2021, December 9). Forest Fire Prediction with Artificial Neural Network (Part 1). Retrieved from https://medium.com/brandon-lammey-intro-to-ai/forest-fire-prediction-with-artificial-neural-network-part-1-991ef74bbd5c

[3] IJRASET Publication. (2020, January 1). Forest Fire Prediction System using Machine Learning. Retrieved from https://www.academia.edu/44767266/Forest_Fire_Prediction_System_using_Machine_Learning

[4] Gulati, A. P. (2022, August 24). Forest Fire prediction using Machine Learning. Retrieved from https://www.analyticsvidhya.com/blog/2021/10/forest-fire-prediction-using-machine-learning/

[5] Saponara, S., Elhanashi, A. & Gagliardi, A. Real-time video fire/smoke detection based on CNN in antifire surveillance systems. J Real-Time Image Proc 18, 889–900 (2021).

[6] Qingjie Zhang, Jiaolong Xu, Liang Xu and Haifeng Guo, Deep Convolutional Neural Networks for Forest Fire Detection. IFMEITA 2016.

[7] Pragati, S. S. (2019-2020). International Journal Of Advance Scientific Research. Forest Fire Detection Using Machine Learning, 1,2.

[8] Prof. K.Angayarkkani, D. (2009). IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.3,. Efficient Forest Fire Detection System: A Spatial Data Miningand Image Processing Based Approach.

[9] Pragati, S. S. (2019-2020). International Journal Of Advance Scientific Research. Forest Fire Detection Using Machine Learning, 1,2.

[10] A Arul, R. S. (2021, May). Fire Detection System Using Machine Learning. Retrieved from ResearchGate: https://www.researchgate.net/publication/351926970_Fire_Detection_System_Using_Machine_Learning.

[11] Eric den breejen, M. B. (1998, November). FOREST FIRE DETECTION USING MACHINE LEARNING. Retrieved from Researchgate: https://www.researchgate.net/profile/Klamer-Schutte/publication/2478027_Autonomous_Forest_Fire_Detection/links/0912f514831a07aee6000000/Autonomous-Forest-Fire-Detection.pdf