A PROJECT REPORT ON

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

TEAM ID **:** PNT2022TMID21021
COLLEGE NAME : Sri Sairam Institute of Technology
DEPARTMENT : Electronics and communication Engineering
DOMAIN : Climate change
TECHNOLOGY : Artificial intelligence

Submitted By

TEAM MEMBERS:

LEAD: YASWANTH KUMAR YENDULURI - 412419106096
SARAN R -412419106072
JEREMIN S.H -412419106035
SURESHKUMAR S -412419106087
VARADHARAJ K -412419106092

Introduction:

1.1Project overview:

Fire can make major hazards in this hectic world. All buildings and vehicles used in public transportation have fire prevention and fire protection systems due to the accelerated number in the fire incidents. Also, many of the firms conduct a mock fire drill in every occurrence of months to protect their employees from the fire. This would help them to understand what to do or what not to do when a fire situation happens. Forests are one of the main factors in balancing the ecology. It is very harmful when a fire occurs in a forest. But most of the time, the detection of forest fire happens when it spread over a wide region. Sometimes, it could not be possibleto stop the fire. As a result,the damage of the environment is higher than predictable. The emission of large amount of carbon dioxide(CO_2) from the forest fire damages the environment. As well as it would lead to complete disappearance of rare species in the world (Alkhatib, 2014). Also, it can make an impact on the weather, and this make major issues like earthquakes, heavy rains, floods and so on.

A research study shows an automatic fire detection can be divided into three groups: aerial, ground and borne detection. The ground-based systems use several staring black and white video cameras are used in fire detection which detect the smoke and compares it with the

natural smoke. The main benefit of using this system is high temporal resolution and spatial resolution. So that, the detection is easier (Eric den breejen,1998). But these mechanisms still have some drawbacks in detecting the early stage of the fire. So that, it is highly important to introduce a system to detect the fire early as possible.

1.2 Purpose:

We propose a novel system for detecting fire using Convolutional Neural Networks (CNN). Detection of fire can be extremely difficult using existing methods of smoke sensorsinstalled in the buildings. They are slow and cost inefficient due to their primitive design and technology. This paper critically analyzes the scope of Artificial intelligence for detection and sending alerts with video from CCTV footages. This project uses self-built datasetcontaining video frames with fire. The data is then preprocessed and use the CNN to build a machine learning model. The test set of the datasetis given as input for validating the algorithm and experiments are noted. The project focus on building cost efficient and highly accurate machine that can be used in almost any usecase of fire detection.

LITERATURE SURVEY:

Existing Problem:

The existing system for detectingfire are smoke alarms and heat alarms.The main disadvantage of the smoke sensor alarm and heat sensor alarms are that just one module is not enough to monitor all the potential fire prone places. The only way to prevent a fire is to be cautious al the time.Even if they are installedin every nook and corner,it just is not sufficient for an efficient output consistently. As the number of smoke sensor requirement increase the cost will also increase to its multiple.The proposed system can produceconsistent and highly accurate alerts within seconds of accident of the fire. It reduces cost because only one software is enough to power the entire network of surveillance. Research is active on this field by data scientists and machine learning researchers. The real challenge is to minimize the error in detection of fire and sending alerts at the right time.

References:

1. Saponara, S., Elhanashi, A. & Gagliardi,A. Real-time video fire/smoke detectionbasedon CNN in antifire surveillance systems. J Real-Time Image Proc 18, 889–900 (2021).

2. Qingjie Zhang,Jiaolong Xu, Liang Xu and Haifeng Guo, Deep Convolutional NeuralNetworks for Forest Fire Detection. IFMEITA 2016.

3. Pragati, S. S. (2019-2020). International Journal Of Advance ScientificResearch. ForestFireDetection Using Machine Learning,

[
4] A Arul, R. S. (2021, May). Fire DetectionSystem Using MachineLearning. Retrievedfrom ResearchGate: https://www.researchgate.net/publication/351926970_Fire_Detection_System_Using_Machine_Learning.

1. Yuanbin Wang, L. D. (2019). Journalof algorithms and Computational technology.Forest fire image recognition based on convolutional neural network, 1.

2. Kim, Byoungjun,and Joonwhoan Lee. 2019. "AVideo-Based Fire DetectionUsing DeepLearning Models" *Applied Sciences* 9, no. 14: 2862

1. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. https://doi.org/10.3390/f12020217

2. BoWFire Dataset. Available online: https://bitbucket.org/gbdi/bowfire-dataset/downloads/(accessed on 1 January 2021). 29.

3. FD-Dataset. Available online: http://www.nnmtl.cn/EFDNet/ (accessed on 1 January2021).

4. Forestry Images. Available online: https://www.forestryimages.org/browse/subthumb.cfm?sub=740 (accessed on 1 January2021). 31. [10].

Problem Statement:

In earlier times fires were detected with the help of watching towers or using satellite images.

- Satellites collect images and send it to the monitoring authority which will decide by seeingimages that it is a fire or not.
- However, this approach was prolonged as the fire may have spread in the large areas and caused somuch damage before the rescue team came.
- In the watching tower method, there was a man always standing on the tower who would monitor the area and inforM if there was fire.
- This method was also slow because before the man got to know about the fire it may have spread in the inner parts of forest, also it always requires a man who must be present there.
- Since, we know that some areas, especially forest areas are large so it is practically impossible to put a man in every part of forest from where they can monitor the forest area.
- So, both these approaches of watching towers and satellite images failed to detect fire as early as possible to reduce the damage done by fire Problems in fire detection:

- There were mainly two problems in fire detection as discussed:

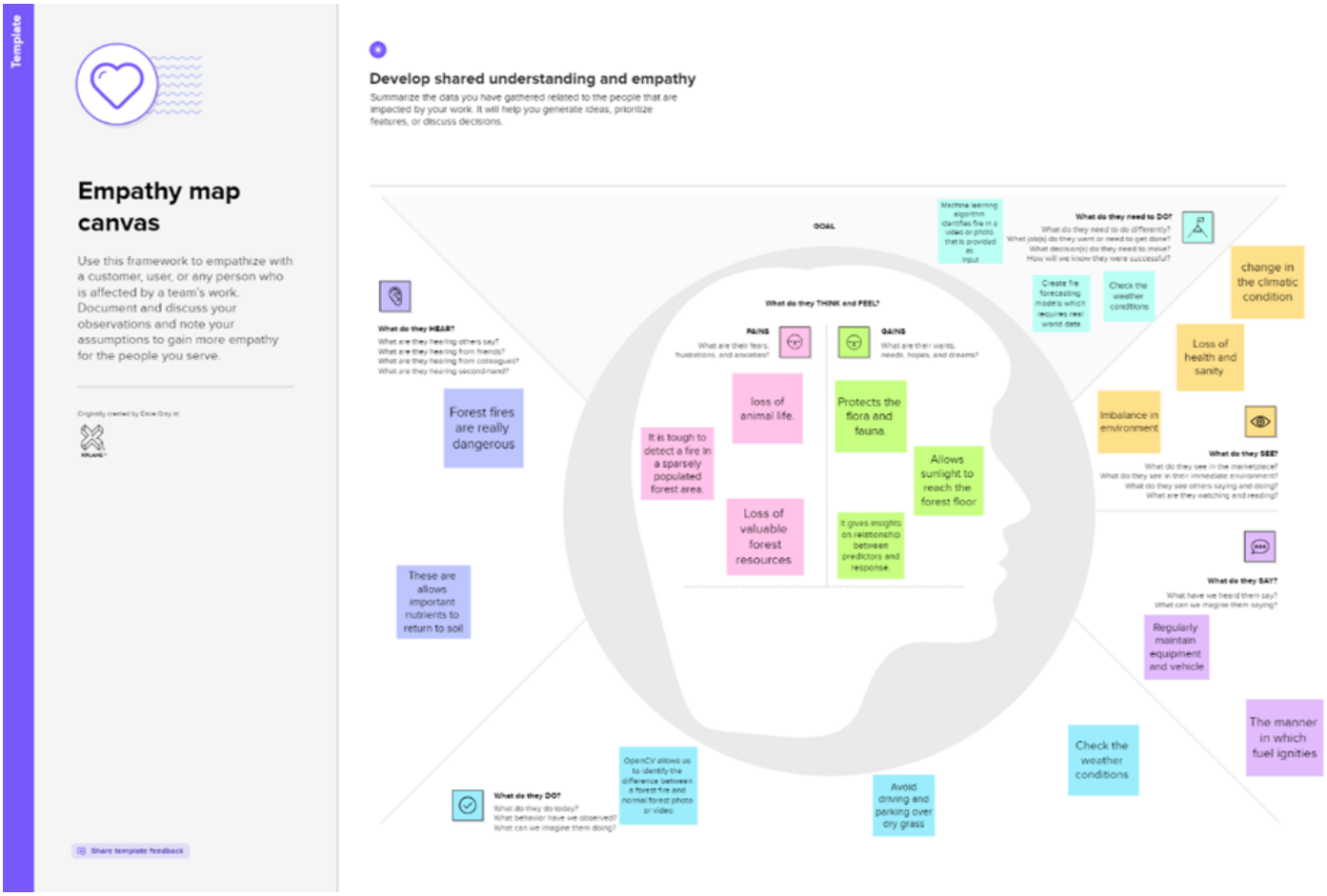
- (a). Judging criteria for the fire: Edge is set, on the off chance that the worth is morenoteworthy than edge, it is a fire, else not.
- So, this problem was removed by using machine learning techniques by many researchers.
- (b). Connection of nodes: Traditional systems used cables to connect alarm with thedetectors.
- Cable was mainly of copper. But copper wire may be costly or it can suffer from fault in the mid-way.

- So, this problem was removed using wireless sensor networks.

- So, with the advancement in technology researchers find an efficient method to detectforest fire with the help of Wireless SensorNetwork.
- Fire can be identified by conveying sensor hubs in timberland regions by which theyilluminate about fire.
- Conveying sensor hubs in the timberland regions means placing sensors in every part of the forest and mostly in the prone areas where risk of 9 catching fire is more. With the use of wireless sensor networks,now it is easy to detect the fire in large areas as soon as possible.

Ideation & Proposed Solution:

Empathy Map Canvas:



Ideation & Brainstorming:

Proposed solution:

Proposed Solution Template:

Project team shall fill the followinginformation in proposedsolution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Detecting forestfires in an early stageto avoidmassive damage.
2.	Idea / Solution description	Identifying huge forest fires in real-time utilising AI algorithms withcamera and satellitefootage. The systems thennotify dispatchers and local authorities about thenew ignition
3.	Novelty / Uniqueness	Convolutional NeuralNetwork system allowsusto deliver information more quickly and accurately. It is possible to deploy a comprehensive coverage, which is nearly impossible.
4.	Social Impact / Customer Satisfaction	Monitoring of the potential dangerregions and early identification of fire can greatly minimise the response time, as wellas potential damageand firefighting expenses, while also saving many lives.
5.	Business Model (Revenue Model)	Subscription Model
6.	Scalability of theSolution	Despite the physical distance between resources and users,its regionally scalable system maintains its usability and utility;

REQUIREMENT ANALYSIS :

FUNCTIONAL REQUIREMENT:

SYSTEM	13
HARD DISK	: 500
RAM	: 4gb DDR2
BOARD	: LG 104 KEYS KEYBOARD
MOUSE	: LOGITECH
MONITOR	: 15INCH TFT COLOR MONITOR

NON-FUNCTIONAL REQUIREMENT :

Software Specifications Google Colaboratory — Colaboratory is a free Jupyter notebook environment provided by Google where one can use free GPUs and TPUswhich requires no setup and runs entirely in the cloud. The Jupyter Notebook is an open-source web application which allows to create and share documents that contain live code, equations, visualizations and narrative text[11]. A notebook is a list of cells. Cells contain either explanatory text or executable code and its output. With Colaboratory one can write and execute code, save and share their analyses, and access powerful computing resources.

PROJECT DESIGN:

DATA FLOW DIAGRAM

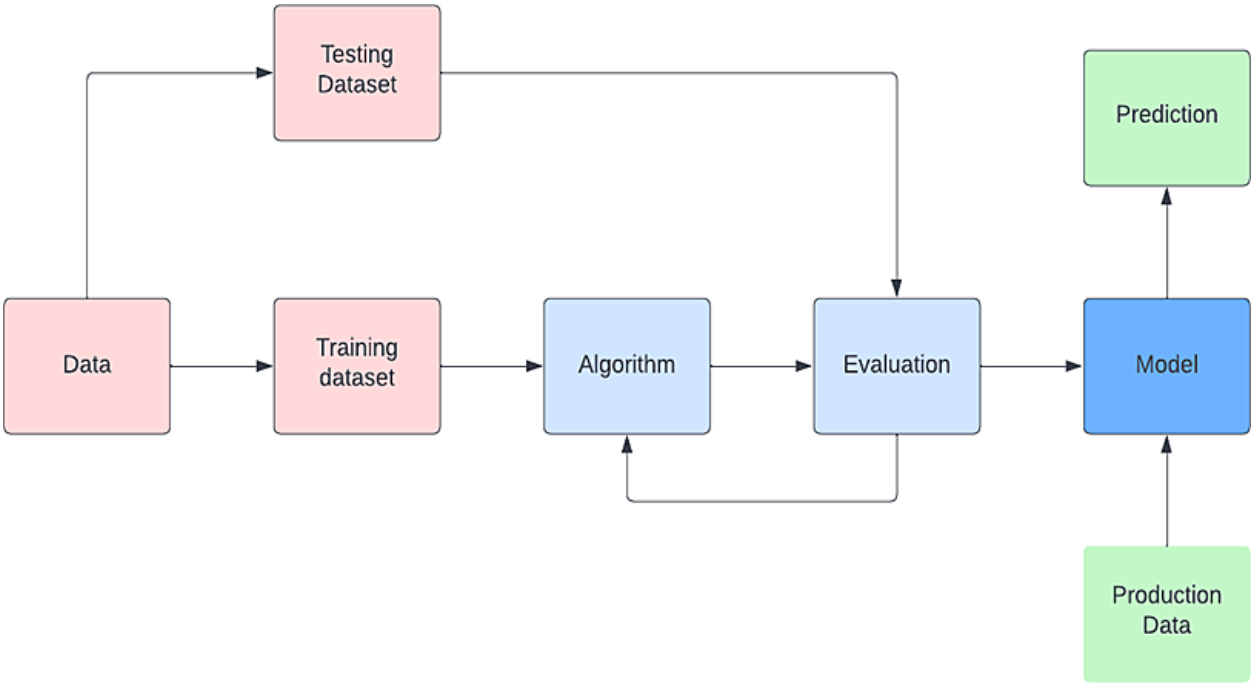
DataFlow Diagrams:

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

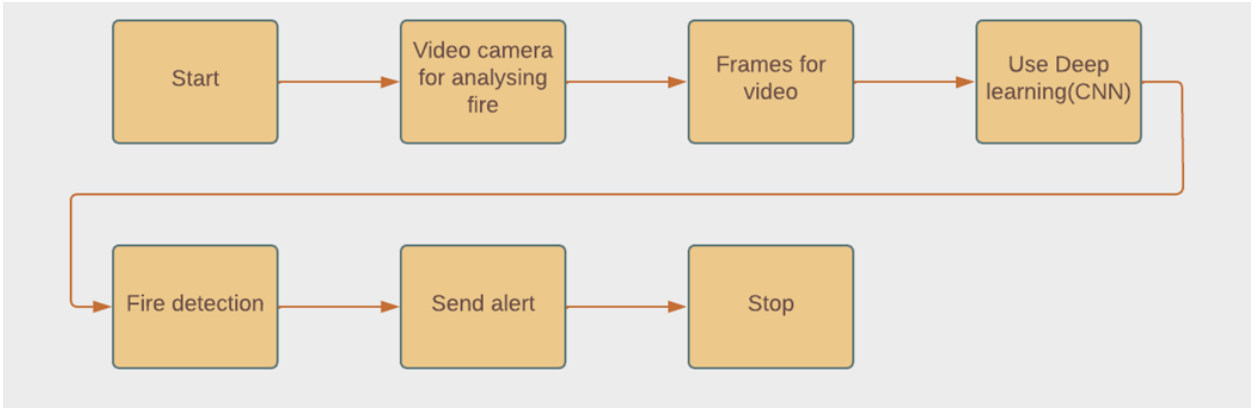
SIMPLIFIED DATA:

1. COLLLECT DATA
2. EVALUATE DATASET

3. IMPLEMENT ALGORITHMS
4. EVALUATE THE ACCURACY FOR EACH ALGORITHMS
5. DISPLAY THE REUSLTS



IndustryStandard DFD:



User Stories:

User type	Functional Requireme	User story	User Story/Task	Acceptance criteria	Priority	Release

	nt	numb er				
Environmentali st	Collect the data	USN -1	Environmentalis ts help the publicmake informed decisions about the useof limited naturalresource s. They do research, producereports, write articles, lecture, issue press releases, lobby congress, fundraise, and campaign. The daily routinedepends on the specialty.	It is necessary to collect the right data else the prediction maybecome wrong.	High	Sprint – 1
		USN - 2	Identify algorithms that can be used forprediction.	To collect thealgorithm to identify the accuracy level of each algorithm.	Medi um	Sprint - 2
	Implement algorithm	USN - 3	Identify the the accuracy of all algorithms that are beingused.	Accuracy of each algorithm is calculated so that it is easy to obtain themost accurate output.	High	Sprint – 2

		USN - 4	Evaluate theDataset.	Data is evaluated before processing.	Medi um	Sprint – 1
	Evaluate accuracy of algorithm	USN - 5	Identify a accuracy, nd precision recall of each algorithm.	These a values re importa for nt rig obtaini ht ng the output.	High	Sprint – 3
	Display results	USN - 6	Outputs are fromeach algorithm obtained.	It is highly used to predict the effect and to take precautionary measures	High	Sprint - 4

TECHNOLOGY ARCHITECTURE:

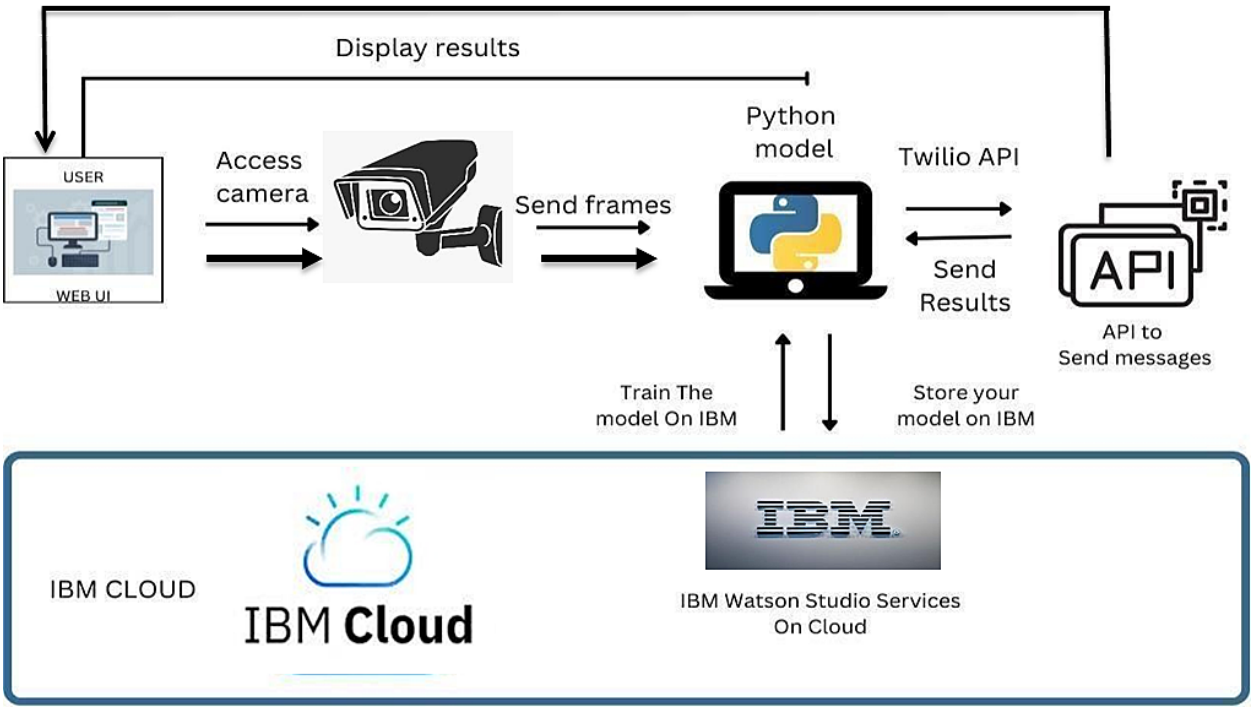


Table-1:

Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.sensor based system,application based system	Python/HTML ,CSS , Java script andreact Js
2.	Input	Video Feed	Web Camera/Video on a site
3.	Application Logic-1	Logic for a process in the application e.g: registration process.	Python
4.	Application Logic-2	Logic for a process in the application e.g registration	Python

		process successful then go to login process	
5.	Dataset	Using Test setand Train set,train the model	Data set from Cloud Storage, Database
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	ture (Server / Cloud),API	Application Deployment on Local System /Cloud Local,Cloud Server Configuration.	undry, Kubernetes, etc.

Table-2:

Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	open-source frameworks used.	Technologyof Opensource framework
2.	Security Implementatio ns	Mandatory Access Control (MAC)and Preventative Security Control is used	e g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	scalability of architecture e g: early alarm in realtime whenthe forest	Technology used

		occurs.	
4.	Availability	availability of application (e g. customer serviceand support to the mobile)	Technology used
5.	Performance	Enhance the performance by using IBM CDN	IBM Content Delivery Network

Solution Requirements (Functional & Non-functional)

Functional Requirements:

Following are the functional requirements of the proposedsolution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**Classification**	There are four classifications of fire cause: accidental, natural, incendiary, and undetermined
FR-2	**Tagging**	It is a classification taskwith a higherdegree ofprecision. It helps to identify several objects within an image.It tags andtracks animals and forestsituations
FR-3	**Localization**	The localisation of the node would be done usingsatellite communication to reduce coverage holes and ensure maximum range with the least latency. This node would communicate data to a monitoring station with its location and send alerts according to the sensed thresholds breached based on the novel logic algorithm.

FR-4	**Detection**	The system, using Moderate Resolution Imaging Spectro-Radiometer (MODIS), Advanced Very HighResolution Radiometer (AVHRR), and Spinning Enhanced Visible and Infra Red Imager (SEVIRI) data,provides near real-time integrated information about both the fire presence and dangerover the affected area
FR-5	**Semantic Segmentation**	Semantic segmentation describes the process ofassociating eachpixel of an image witha class label. It includes Sentinel-1, Sentinel -2, Sentinel-3and MODIS.
FR-6	**Instance Segmentation**	In Instance Segmentation, bounding boxesare generated for each instance of multiple categories present alongwith the objectsegmentation masks. It includes Hydrology, Rivers, Lakes and Audioweather conditions.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Many methods have been proposed to detect forest fires, such as camera-based systems, WSN-based systems, and machine learning application-based systems, with both positive and negative aspects andperformance figures of detection.
NFR-2	**Security**	As this processis designed with aminimum delay, the fire can be detected within the initial stage, and the responsible parties can takenecessary actions in a shorter period, which willminimize the damage. This ensures security ofwell beings.

NFR-3	**Reliability**	The system shall be supervised either electrically or with satellite or even by software-directed polling of field.The panel, detectors and modules shall preferably used.
NFR-4	**Performance**	In the event of a fire, the primary objective of using drones is to gather situational awareness, which can be used to direct the efforts of the firefighters in locating and controlling hot spots. Just like urbanfires, forest firesto require monitoring so that firefighters know what they are dealingwith. Model willachieve high accuracy
NFR-5	**Availability**	By making field testing,Threshold ratio analysis it ensures minimum up time and performance
NFR-6	**Scalability**	A widely used measure of fire intensity is fireline intensity, which is the rate of heat transfer per unit length of the fire line (measured in kW m−1) and represents the radiant energy release in the flaming front,

PROJECT PLANNING&ESTIMATION

SPRINT PLANNING & ESTIMATION &
SPRINT DELIVERY SCHEDULE

Product Backlog,Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement	User storynumber	User Story/Task	Acceptance criteria	Priority	Team Members

Sprint – 1	Collect the data	USN -1	Environmentalists help the publicmake informed decisions about the use of limited naturalresources. They do research, produce reports, write articles, lecture, issue pressreleases, lobby congress, fundraise,and campaign. The daily routinedepends on the specialty.	It is necessary to collect the right data else the prediction maybecome wrong.	High	• Saran • Varadharaj • Yaswanth • Jeremin • Suresh
Sprint - 2	Image preprocessing	USN - 2	Identify algorithms that can be used forprediction.	To collectthe algorithm to identify the accuracy level of each	Medium	

				algorithm.		
Sprint – 2	Implement algorithm	USN - 3	Identify theaccuracy of allthealgorithms that are being used.	Accuracy of each algorithm is calculated so that it is easy to obtain the most accurate output.	High	

Sprint – 1	Reviewing themodel	USN - 4	Evaluate theDataset.	Data is evaluated before processing.	Medium	● Saran ● Varadharaj ● Yaswanth ● Jeremin ● Suresh

Sprint – 3	Evaluate accuracy of algorithm	USN - 5	Identify accuracy, precision andrecall of each algorithm.	These valuesare importa nt for obtaining the right output.	High	
Sprint - 4	Display results	USN - 6	Outputs fromeach algorithm areobtained.	It is highly used to predict the effect andto take precautionary measures	High	

Project Tracker, Velocity& Burndown Chart:

Sprint	Total Story Points	Duration	Sprint StartDate	Sprint End Date (Planned)	Story PointsCompleted (as on Planned EndDate)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct2022	29 Oct2022	8	29 Oct2022
Sprint-2	20	6 Days	31 Oct2022	05 Nov 2022	7	08 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	8	15 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	7	20 Nov 2022

Project Tracker, Velocity & Burndown Chart:(4 Marks) **Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV)per iteration unit (story points per day)

AV = Sprint duration /velocity = 7/10 =0.7

CODING & SOLUTIONING

FEATURE:

"""Applying ImageDataGenerator functionality To Testset And Trainset.ipynb

Automatically generated by Colaboratory.

Original file is located at
https://colab.research.google.com/drive/13PPU6URoKuCkNiC_SciYaXrxAfuWTDMZ

Team ID: PNT2022TMID21021

Applying ImageDataGenerator functionality To Testset And Trainset

```
"""

import keras
from keras.preprocessing.image import ImageDataGenerator

#Defining the parameters/arguments for ImageDataGenerator class
train_datagen =
ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_rang
e=0.2,horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

#Applying the ImageDataGenerator function to the trainset
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM/Dataset/Datas
et/train_set',target_size=(128,128),batch_size=32,class_mode='binary')

#Applying ImageDataGenerator functionality to the testset
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IBM/Dataset/Datase
t/test_set',target_size=(128,128),batch_size=32,class_mode='binary')
```

```
# -*- coding: utf-8 -*-
"""OpenCv_for_Video_processing.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1Ae3ZYf_2zodEVMaxnMFxoEu7AJRBmIvp

Team ID: PNT2022TMID21021
```

```
OpenCv for Video processing
"""

#importing required libraries
!pip install tensorflow
!pip install opencv-python
!pip install opencv-contrib-python
import tensorflow as tf
import numpy as np
from tensorflow import keras
import os
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image

train=ImageDataGenerator(rescale=1./255,
                         shear_range=0.2,
                         rotation_range=180,
                         zoom_range=0.2,
                         horizontal_flip=True)
train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)

train_dataset =
train.flow_from_directory("/content/drive/MyDrive/IBM/Dataset/Dataset/train_set",
target_size=(128,128), batch_size = 32,
                                 class_mode = 'binary' )

test_dataset =
test.flow_from_directory("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set",
target_size=(128,128), batch_size = 32,
                                 class_mode = 'binary' )

test_dataset.class_indices
```

```
#to define linear initialisation import sequential
from keras.models import Sequential
#to add layer import Dense
from keras.layers import Dense
#to create convolution kernel import convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')

model = keras.Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

model.add(Dense(150,activation='relu'))

model.add(Dense(1,activation='sigmoid'))

model.compile(loss = 'binary_crossentropy', optimizer = "adam", metrics = ["accuracy"])

r = model.fit(train_dataset, epochs = 5, validation_data = test_dataset)

pred = model.predict(test_dataset)
pred = np.round(pred)
```

```
pred

print(len(pred))

model.save("/content/forest1.h5")

#import load_model from keras.model
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2

model = load_model("/content/forest1.h5")

import matplotlib.pyplot as plt
plt.plot(r.history['loss'],label='loss')
plt.plot(r.history['val_loss'],label='val_loss')
plt.legend()

plt.plot(r.history['accuracy'],label='acc')
plt.plot(r.history['val_accuracy'],label='val_acc')
plt.legend()

def predictImage(filename):
  img1=image.load_img(filename,target_size=(128,128))
  plt.imshow(img1)
  y=image.img_to_array(img1)
  x=np.expand_dims(y,axis=0)
  ctr=model.predict(x)
  print(ctr)
  if ctr==1:
```

```
    plt.xlabel("Fire detected",fontsize=30)
  elif ctr==0:
    plt.xlabel("No fire detected",fontsize=30)

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/forest/2017_10_12_09_01_56.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with fire/FORESTFIRE (1).jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/forest/beautiful_mountain_forest_wallpaper_1920x1200.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with fire/ring_of_fire_bailey_colorado_rocky_mountain_forest_wildfire_picture_id157384116.jpg")

pip install twilio

pip install playsound

pip install opencv-python

#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
```

```
#load the saved model
model = load_model(r'/content/forest1.h5')
#define video
video = cv2.VideoCapture('/content/drive/MyDrive/forest fire.mp4')
#define the features
name = ['forest','with forest']

"""Predictions.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1PVTQugaCBW35ByIHoQO2bbNg8ZPyyl89

Team ID: PNT2022TMID21021

Predictions
"""

import keras
from keras.preprocessing.image import ImageDataGenerator

#Defining the parameters/arguments for ImageDataGenerator class
train_datagen =
ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_rang
e=0.2,horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

#Applying the ImageDataGenerator function to the trainset
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM/Dataset/Datas
et/train_set',target_size=(128,128),batch_size=32,class_mode='binary')

#Applying ImageDataGenerator functionality to the testset
```

```
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IBM/Dataset/Dataset/test_set',target_size=(128,128),batch_size=32,class_mode='binary')

#import model building libraries

#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten

import warnings
warnings.filterwarnings('ignore')

#initializing the model
model=Sequential()

#add convolutional layer
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add flatten layer
model.add(Flatten())

#add hidden layer
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid'))

#configure the learning process
```

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])

#Training the model using 'fit' method
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,val
idation_steps=4)

#saving the model
model.save("forest1.h5")

#import load_model from keras.model
from keras.models import load_model
#import image class from keras
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2

#load the saved model
model = load_model("forest1.h5")

img=image.load_img('/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with
fire/Bandipur_fires_2019.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)

pred=model.predict(x)
```

"""Train_image_classification_Model.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1YDGFKlN1CzQrYucsnSxHSlYiD36KOJCn

Team ID: **PNT2022TMID21021**
"""

```
!pip install tensorflow
!pip install opencv-python
!pip install opencv-contrib-python

pip install opencv-python

import tensorflow as tf
import numpy as np
from tensorflow import keras
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image

train=ImageDataGenerator(rescale=1./255,
                         shear_range=0.2,
                         rotation_range=180,
                         zoom_range=0.2,
                         horizontal_flip=True)
train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)

train_dataset =
train.flow_from_directory("//content/drive/MyDrive/IBM/Dataset/Dataset/test_set",
                          target_size=(128,128),
                          batch_size = 32,
```

```
                                class_mode = 'binary' )

test_dataset =
test.flow_from_directory("///content/drive/MyDrive/IBM/Dataset/Dataset/train_set",

                                target_size=(128,128),
                                batch_size = 32,
                                class_mode = 'binary' )

test_dataset.class_indices

#to define linear initialisation import sequential
from keras.models import Sequential
#to add layer import Dense
from keras.layers import Dense
#to create convolution kernel import convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')

model = keras.Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

model.add(Dense(150,activation='relu'))

model.add(Dense(1,activation='sigmoid'))

model.compile(loss = 'binary_crossentropy',
          optimizer = "adam",
```

```
            metrics = ["accuracy"])

r = model.fit(train_dataset, epochs = 5, validation_data = test_dataset)

predictions = model.predict(test_dataset)
predictions = np.round(predictions)

print(len(predictions))

model.save("/content/forestmodel.h5")

#import load_model from keras.model
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2

model = load_model("/content/forestmodel.h5")

import matplotlib.pyplot as plt
plt.plot(r.history['loss'],label='loss')
plt.plot(r.history['val_loss'],label='val_loss')
plt.legend()

plt.plot(r.history['accuracy'],label='acc')
plt.plot(r.history['val_accuracy'],label='val_acc')
plt.legend()

def predictImage(filename):
  img1=image.load_img(filename,target_size=(128,128))
  plt.imshow(img1)
  y=image.img_to_array(img1)
  x=np.expand_dims(y,axis=0)
  val=model.predict(x)
  print(val)
  if val==0:
    plt.xlabel("No fire Detected!!",fontsize=30)
```

```
  elif val==1:
    plt.xlabel("Fire Detected!!",fontsize=30)

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with
fire/19464620_401.jpg")

predictImage("////content/drive/MyDrive/IBM/Dataset/Dataset/test_set/forest/0.64133000_151
9374442_forest_deep.jpg")

pip install twilio

pip install playsound

pip install opencv-python

#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound

#load the saved model
model = load_model(r'/content/forestmodel.h5')
#define video
video = cv2.VideoCapture('//content/drive/MyDrive/IBM/forest video')
#define the features
name = ['forest','with forest']

video.isOpened()

from tensorflow.keras.preprocessing import image

from IPython.display import Audio
```

```
while(video.isOpened()):
  success,frame=video.read()
  cv2.imwrite("with fire/19464620_401.jpg",frame)
  img=image.load_img("//content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with
fire/19464620_401.jpg",target_size=(128,128))
  x=image.img_to_array(img)
  x=np.expand_dims(x,axis=0)
  pred=model.predict(x)
  p=pred[0]
  print(pred)
  cv2.putText(frame,"predicted class = ",(100,100),cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
  if pred[0]==1:
    account_sid = 'AC9f6fc46b088bc81bbeef4a802cd06864'
    auth_token = '29f7d63c9d6ce55afbecb683a85072c2'
    client=Client(account_sid,auth_token)
    message=client.messages \
    .create(
        body="Forest fire is detected ,stay alert",
        from_='+15095193634',
        to='+919491291847')
    print(message.sid)
    print('Fire detected')
    print('SMS sent')
    wn=Audio('/content/tornado-siren.mp3',autoplay=True)
    display(wn)
    break
  else:
    print('No danger')
    break
  if cv2.waitKey(1) & 0xFF==ord('a'):
    break
video.release()
cv2.destroyAllWindows()
```

"""Final_Code.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1x-IftWPDybimjmOL08Xrw-D1F6FAZ3VM

Emerging Methods for Early Detection of Forest Fires
Team ID: PNT2022TMID21021

Image Pre-Proecessing

1.Importing the ImageDataGenerator Library
"""

```
import numpy as np
import keras
from sklearn.model_selection import train_test_split
from keras.models import Sequential, load_model
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard
from keras.callbacks import ReduceLROnPlateau
from keras.layers import Conv2D, Dropout, Dense, Flatten, MaxPooling2D, SeparableConv2D,
Activation, BatchNormalization
import matplotlib.pyplot as plt
import time
import os
import tensorflow as tf
```

"""**2.Define parameters for ImageDataGenerator Class**"""

```
train_datagen=ImageDataGenerator(rescale=1./255,
                                 shear_range=0.2,
                                 rotation_range=180,
                                 zoom_range=0.2,
                                 horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

"""**3.Applying ImageDataGenerator Functionality to Trainset and Testset**

a. for dataset
"""

x_dataset
=train_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM/Dataset/Dataset",target_size
= (128,128), class_mode = "binary",batch_size = 32)

"""b. for trainset"""

x_train
=train_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM/Dataset/Dataset/train_set",ta
rget_size = (128,128), class_mode = "binary",batch_size = 32)

"""c. for testset"""

x_test
=test_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM/Dataset/Dataset/test_set",tar
get_size = (128,128), class_mode = "binary",batch_size = 32)

x_train.class_indices

"""# **Model Building**

1. Importing Model Building libraries
"""

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')

"""**2. Initializing the model**"""

model=Sequential()

"""**3. Adding CNN layers**

a. Adding Convolution layers

"""

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

"""*b. Adding pooling layer*"""

model.add(MaxPooling2D(pool_size=(2,2)))

"""*c. Adding Flatten layer*"""

model.add(Flatten())

model.summary()

"""**4. Adding Dense layers**

a. Adding hidden layers
"""

model.add(Dense(150,activation='relu'))

"""*b. Adding Output layer*"""

model.add(Dense(1,activation='sigmoid'))

"""**5. Configuring the Learning Process**

"""

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

"""**6. Training the model**"""

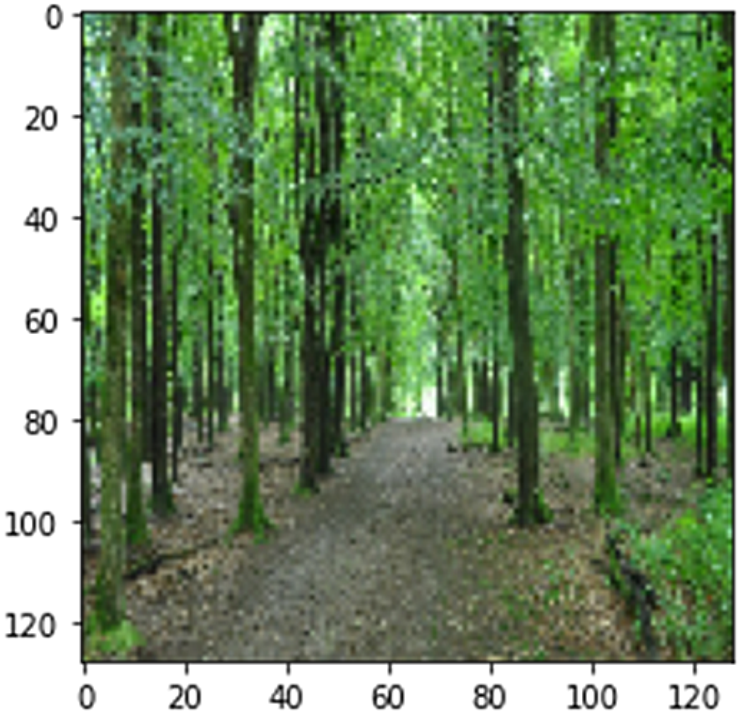
#training using fit method
r=model.fit_generator(x_train,steps_per_epoch=14, epochs=10,validation_data=x_test,
validation_steps=2)

import matplotlib.pyplot as plt
plt.plot(r.history['loss'],label='loss')
plt.plot(r.history['val_loss'],label='val_loss')

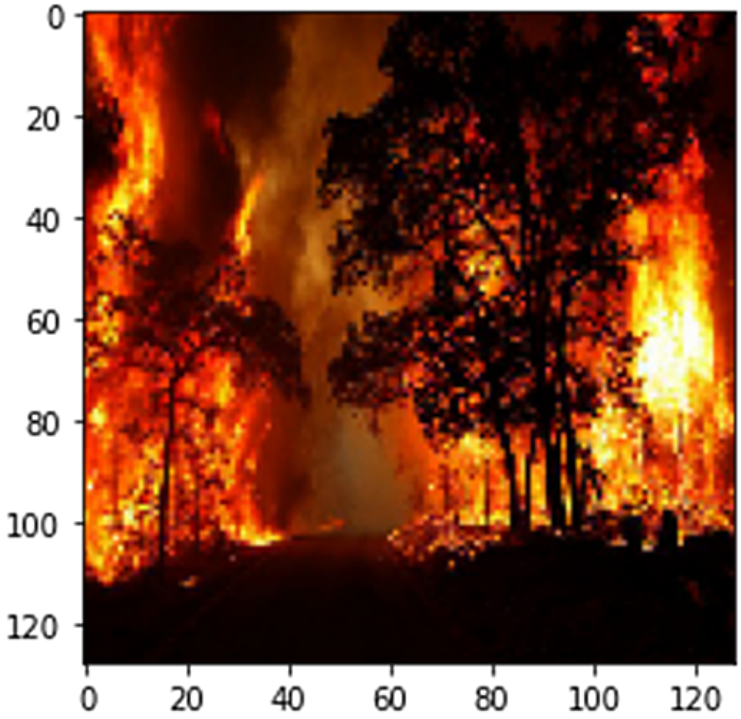
```
plt.legend()

plt.plot(r.history['accuracy'],label='acc')
plt.plot(r.history['val_accuracy'],label='val_acc')
plt.legend()

"""**7. Save the model**"""

model.save("/content/forest_model.h5")

"""**8. Test the model**"""

import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import cv2

model=load_model('/content/forest_model.h5')

img=image.load_img('/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with
fire/Bandipur_fires_2019.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)

"""# **Predictions**"""

pred=model.predict(x)

pred

def predictImage(filename):
  img1=image.load_img(filename,target_size=(128,128))
  plt.imshow(img1)
  y=image.img_to_array(img1)
  x=np.expand_dims(y,axis=0)
  ctr=model.predict(x)
  print(ctr)
  if ctr==0:
    plt.xlabel(" No Fire detected",fontsize=30)
```

```
  elif ctr==1:
    plt.xlabel("Fire detected",fontsize=30)

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/forest/1170x500_Ireland_web.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/forest/01_NeilBurnell_Mystical_photoverticall.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/forest/2017_10_12_09_01_56.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with fire/Forest_fire_MNRF_esize_IMG_6743.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with fire/How_to_Protect_Your_Home_From_Forest_Fire_1024x588.jpg")

predictImage("/content/drive/MyDrive/IBM/Dataset/Dataset/test_set/with fire/uttarakhand_forest_fire_750x500.jpg")

"""# **Sending Alert message**"""

pip install twilio

pip install playsound

pip install opencv-python

#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
```

```
#load the saved model
model = load_model(r'/content/forest_model.h5')
#define video
video = cv2.VideoCapture('/content/drive/MyDrive/forest fire.mp4')
#define the features
name = ['forest','with forest']

from twilio.rest import Client

account_sid = 'AC9f6fc46b088bc81bbeef4a802cd06864'
auth_token = '29f7d63c9d6ce55afbecb683a85072c2'
client = Client(account_sid, auth_token)


message = client.messages.create(
                     messaging_service_sid='MG55c5ca2c99b3b2f04047a3b7fe504a56',
                     body='Forest fire is detected, stay alert!',
                     to='+918610271176'
                   )

print(message.sid)
print("Fire detected")
print("SMS Sent!")
```



No Fire detected



Fire detected

RESULT:

a. **PERFORMANCE METRICES**

IX PERFORMANCE EVALUATION In this project, as of now, we have worked with two different machinelearning models. We calculated the accuracy of these models.The comparison of these models is as follows: SVM Decision Tree Accuracy 0.62 0.99 Precision0 0.54 0.98 Precision 1 0.76 1 Recall 0 0.78 0.99 Recall 1 0.51 0.98 Based on these observations after our experiment and analysis we can clearlycompare the performance ofthe models to predict the chances of fire.

2. ADVANTAGE AND DISADVANTAGE:

Every year it seems like there's another disastrous wildfire in the American West. In 2018, nearly 9 million acres were burned in the US alone. Uncontrolled fires often started accidentally by people, rampage and decimate forests. For most people, a forest fire is synonymous with disaster. But there are some kinds of forest fires that actually benefit theenvironment.

From forests to deserts, wildfires affect air quality, vegetation, human and animal habitats, and climate around the world. Fire managers and researchers are finding ways to use NASA data to battle fires and measure their effects. Burning fires produce both ashes, which falls to the ground like snow but can also get caught up in winds, and smoke, a mixture of gases and particulate matter. These get into the atmosphere and can travel long distances impacting air quality regionally. Wildfires are unplanned fires that start in forests or wildland areas. There are numerous post-fire impacts, including an increase in air pollution and less infiltration of precipitation, contributing to flooding hazards even long after the burn.

A controlled burn is a wildfire that people set intentionally for a specific purpose. Well-thought-out and well-managed controlled burns can be incredibly beneficial for forest management, in part because they can help stop an out-of-control wildfire. The technique is called backburning, and it involves setting a controlled fire in the path of the approaching wildfire. All the flammable material is burnt up and extinguished. When the wildfire approaches, there's no more fuel left for it to keep going, and it dies out. Forest fire science entails understanding how a fire starts, what contributes to the fire and how

the fire might impact future Earth processes. Understanding climatological changes are important to understand how these changes may contribute to fires in thefuture.

Controlled burns are also used to prevent forest fires. Even before human involvement, natural, low-intensity wildfires occurred every few years to burn up fuel, plant debris, and dead trees, making way for young, healthy trees and vegetation to thrive. That new growth in turn supports forest wildlife. Forest managers are now replicating this natural strategy when appropriate, starting manageable, slow-burning fires to make room for the new life that will help keep the forest healthy in the long term.

The same method is one of WWF's strategies for maintaining grassland habitats in the Northern Great Plains. Working with partners such as the U.S. Fish and Wildlife Service, WWF has intentionally burned hundreds of acres of prairie land to revitalize these keyhabitats. The fire burns off tall, aggressive vegetation that isn't as hospitable to wildlife, andmakes room for new growth that attracts bison, birds, and prairiedogs.

This doesn't mean all intentional wildfires are good. Many of the fires intentionally set for agriculture and land clearing are at best ill-advised, and at worst devastating. Slash and burn fires are set every day to destroy large sections of forests. Of course, these forests don't just remove trees; they kill and displace wildlife, alter water cycles and soil fertility, and endanger the lives and livelihoods of local communities. They also can rage out of control. In 1997, fires set intentionally to clear forests in Indonesia escalated into one of the largest wildfires in recorded history. Hundreds of people died; millions of acres burned; already at-risk species like orangutans perished by the hundreds; and a smoke and ash haze hung over Southeast Asia for months, reducing visibility and causing acute health conditions.

When scientists think a fire could be the best solution for revitalizing wild areas, WWF brings the right experts to the table to study the situation and come up with a plan. All fire is risky. To minimize that risk as much as possible, controlled burns must be well-considered, well-planned, and ignited and maintained by trained professionals. Fire can be a tool for conservation, but only when used the right way.

Since 1990, “Wildland Fires” across Canada have consumed an average of 2.5 million hectares a year. These fires occur in forests, shrublands and grasslands. Some areuncontrolled wildfires started by lightning or human carelessness. A small number are prescribed fires set by authorized forest managers to mimic natural fire processes that renew and maintain healthy ecosystems. Wildland fires present a challenge for forest management because they have the potential to be at once harmful and beneficial. They can threaten communities and destroy vast amounts of timber resources, resulting in costly losses.
However, wildland fires are a natural part of the forest ecosystem and important in many parts of Canada for maintaining the health and diversity of the forest. In this way, prescribed fires offer a valuable resource management tool for enhancing ecological conditions and eliminating excessive fuel build-up.

Not all wildland fires should or can be controlled. Forest agencies work to harness the force of natural fire to take advantage of its ecological benefits while at the same time limiting its potential damage and costs. This makes fire control strategies a vital component of forest management and emergency management in Canada. Understanding the complex phenomenon of wildland fire begins with understanding the basic physicalaspects of fire and the ecological role of fire in forests and other wildland areas. Increasingly accurate assessments of the fire situation across Canada are now helping land managers use forest science to reducefire risk and optimize the benefits.

CONCLUSION

From this project we came to the conclusion that decision tree has a remarkable accuracy of 99% in predicting fires in forest areas. This reduces the chances of false alarm to a great extent. Our system is able to differentiate various forest fire scenarios, from initial case (no fire) to detection of fire, fairly accurately. It can accurately determine the growth of fire. This will help in early stages of fire detection and help to confine fire to limited areas before much damage occurs. The system will be very effective in preventing occurrence of false alarms. We aim at monitoring the forests without constant human supervision.

FUTURESCOPE

This project carries a broad prospective for future. Moreover it is a need for great research to be done in this field in the coming years. In future, our project can be extended towards finding an efficient way of localization of the fire, gravity of fire,

direction of spread, area burnt and many more. In our experiment, the process of simulation of forest fire was done by burning the dried leaves directly. We could come up with ways to make this simulation more close to actual forest fires. Moreover, we can include the region.