

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

A PROJECT REPORT

Submitted by:

MAHA PRIYA.T(913119106055)

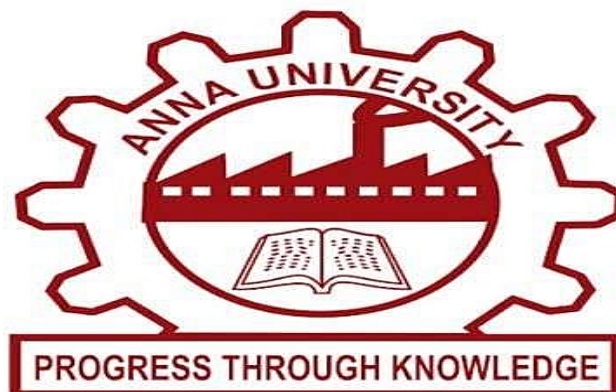
SAKTHI MALAVIKA.U (913119106090)

SUBA SRI.R (913119106112)

YOGA PANDEESWARI.S (913119106122)

TEAM ID :PNT2022TMID23181

**BACHELOR OF ENGINEERING IN ELECTRONICS AND
COMMUNICATION ENGINEERING**



S.NO TABLE OF CONTENT

1 INTRODUCTION

1.1 Project Overview

1.2 Purpose

2 LITERATURE SURVEY

2.1 Existing problem

2.2 Reference

2.3 Problem Statement Definition

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution Fit

4 REQUIREMENT ANALYSIS

4.1 Functional requirements

4.2 Non-Functional requirements

5 PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7 CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Scheme (if applicable)

8 TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9 RESULT

9.1 Performance Metrics

10 ADVANTAGES &DISADVANTAGES

11 CONCLUSION

12 FUTURE SCOPE

13 APPENDIX

Source code

GitHub

Project Demo Link

1.INTRODUCTION

1.1.PROJECT OVERVIEW:

It is difficult to predict and detect forest fires in sparsely populated forest areas and it is more difficult when the prediction is done using ground-based models like cameras. Satellites can be an important source of data prior to and also during the fire due to their reliability and efficiency. The various real time forest fire detection and prediction approaches, results in the goal of informing the local fire authorities.

1.2.PURPOSE:

To detect the forest fire in the early stage. For the early detection of forest fire, the proposed model has an image recognition system method based on Deep learning model.

2.LITERATURE SURVEY

2.1.EXISTING METHOD:

S.NO	AUTHOR	TITLE	NAME OF JOURNAL
1.	Medi Rahul, Karnekanti Shiva, SakethAttiliSanjeet and Nenavath Srinivas Naik.	Early Detection of Forest fire using Deep Learning.	2020.IEEREGION10 Conference(TENCON),2020,pp. 11361140,doi:10.1109/tencon 50793.2020.9293722

- The system involves pre-processing the image data and applying data augmentation such as shearing, flipping, etc.
- It uses models like VGG16 , ResNet50 , and DenseNet121 for the classification of images.
- The model initially divides the train and test sets in 80% and 20% and then sent to the pre-processing phase, where finally it is trained to classify them into two classes

fire and non-fire.

- By using the optimal learning rate the proposed model was able to achieve a training set accuracy of 92.7% and an est set accuracy of 82.57%.

S.NO	AUTHOR	TITLE	NAME OF JOURNAL
2.	Byron Arteaga, Mauricio Diaz, Mario jaoa, University of Naino Pasto Columbia	Deep Learning Applied forest Fire Detection	2020 IEEE International Symposium on signal processing and information Technology(ISSPIT),2020,p p, 16,doi:10,1109/ISSPIT5152 1.2020.9408859

- The data processing was done through open source programming language Python, the cloud service Googlecollab, and deep learning algorithms using Pytorch's library.
- After the data augmentation and pre-processingof the training image, three types of transformation takes place cropping of the image, rotating of an image, and normalizing of the image.
- The classification of images is done by using the pre-trained models of ResNet and VGG pre-trained models.
- To validate the performance of each pre-trained model the k-fold method is used.
- The model obtained during the validationis sent to Raspberry to test its functionality.

S.NO	AUTHOR	TITLE	NAME OF JOURNAL
3.	Raghad k. Mohammed(Depart ment of Basic sciences,college of Density, University Baghdad,Baghdad,Iraq).	A Real-time forest fire and Smoke detection System Using Deep Learning.	International Journal of Nonlinear Analysis and Application 13.1(2022):2053-2063.

- The proposed framework aims to detect smoke and fire based on the images received from the video stream from the Raspberry Pi
- Pre-processing of image data.

- Image data augmentation (Scale, horizontal flip, and vertical flip).
- Pre-training model imagenet dataset ->{inception-ResNet-V2}.
- By fine-tuning the above two steps we have to send that to the fully connected layer with softmax.
- we can view the model accuracy as instead.

S.NO	AUTHOR	TITLE	NAME OF JOURNAL
4.	Suhas.G , Chetan Kumar,Abhishek.B.S, Digvijay Gowda.K.A, Prajwal.R . student of Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore, Karnataka,India	Fire Detection Using Deep Learning.	International Journal of Progressive Research in Science And Engineering Volume-1,Issue-5,August-2020.

- The model is divided into two parts
- a. Data collection and Pre-processing.
- b. Building fire detection model by transfer learning.
- The first step is to gather video frames and it should be divided into two classes fire and non-fire. The collected dataset is divided into train and test sets.
- The second step is to extract the video features of pre-trained models using Keras.
- We have used ResNet-50, Inception V3, and InceptionResNetV2 models to extract the features and various ML algorithms on the extracted features to detect fire in video frames.

2.2.REFERENCES:

- 1.Early detection of forest fire - <https://ieeexplore.ieee.org/document/9293722> using deep learning.
- 2.Deep Learning Applied -<https://ieeexplore.ieee.org/document/9408859> Forest fire Detection.
- 3.A Real-time Forest Fire Smoke detection - https://ijnaa.semnan.ac.ir/article_5899.html System Using Deep Learning.

4.Fire Detection Using -

<https://journals.grdpublications.com/index.php/ijprse/article/view/141>

Deep Learning.

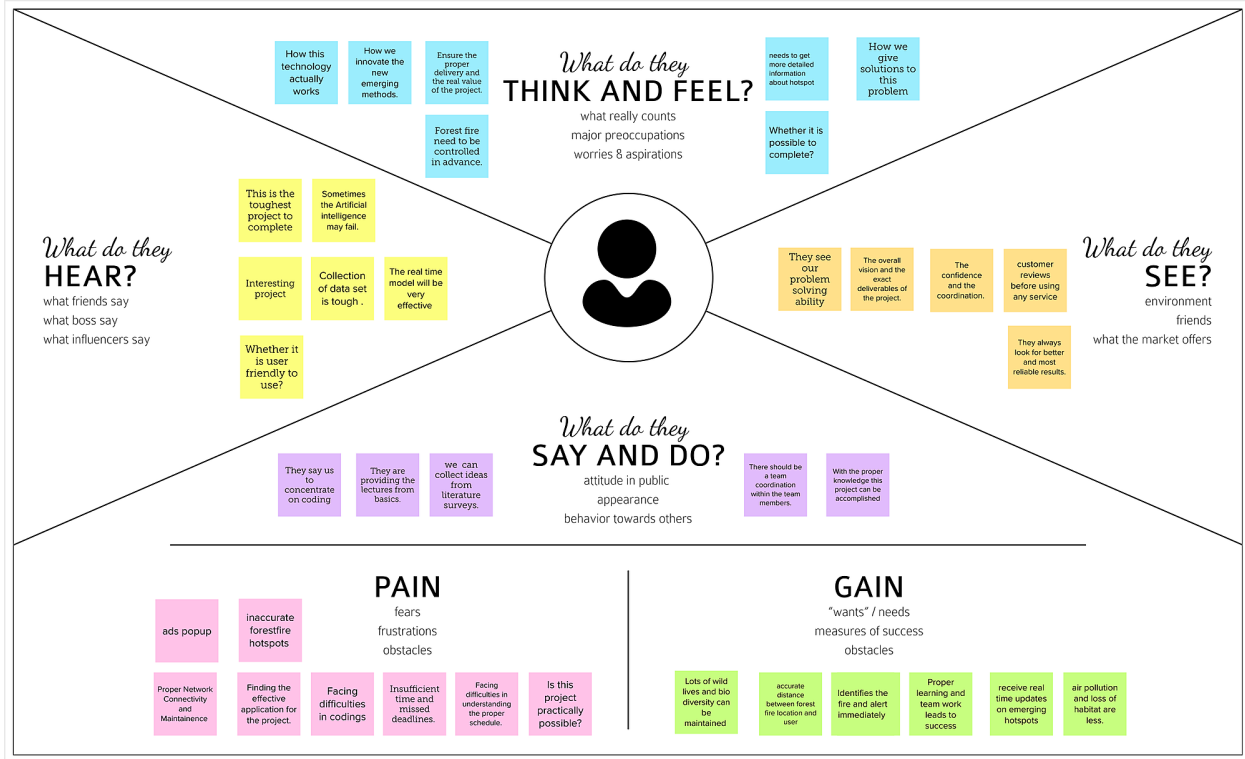
2.3.PROBLEM STATEMENT DEFINITION:

Forest fires is a wide spread and critical factor in the earth's ecosystem. The most effective and vital solution is early detection fires to preserve natural resources and to protect living creatures.

Who does the problem affect?	People living in the forest.
When does the issue occurs?	When there is a climate change in the environment .
Where is the issue occurring?	The issue occurs when there is a difficulty to identify the forest fires.
What is the issue?	Forest fires are a major environmental issue,creating economic and ecological damage while endangering human lives.
Why is it important that we fix the problem?	By solving these issues,it can reduce the forest fire in the beginning stage,by alerting user and can save the ecosystem and human lives.

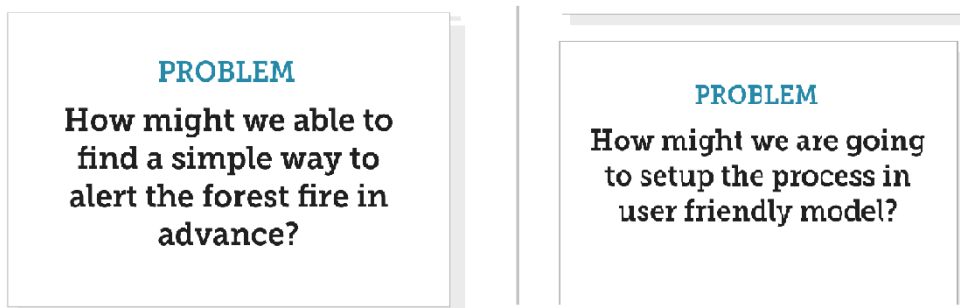
3.IDEATION & PROPOSED SOLUTION

3.1.EMPATHY MAP CANVAS:



3.2.BRAINSTORMING :

Problem Statements:

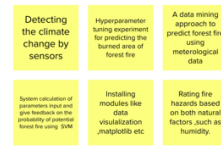


Brainstorm:

Sakthi Malavika.u



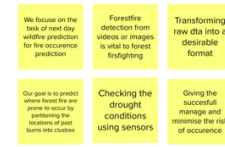
Yoga Pandeewari.S



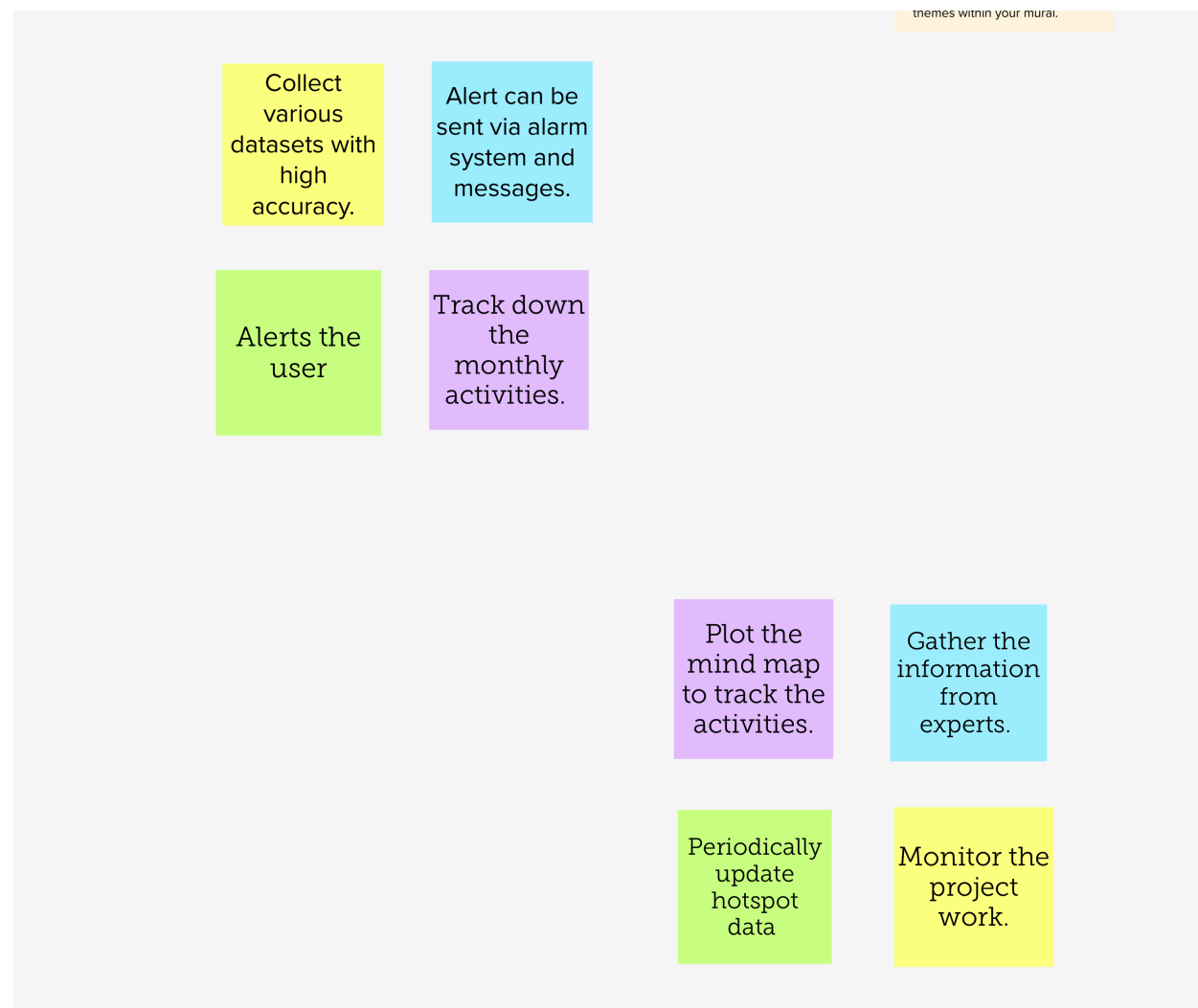
Suba Sri.R



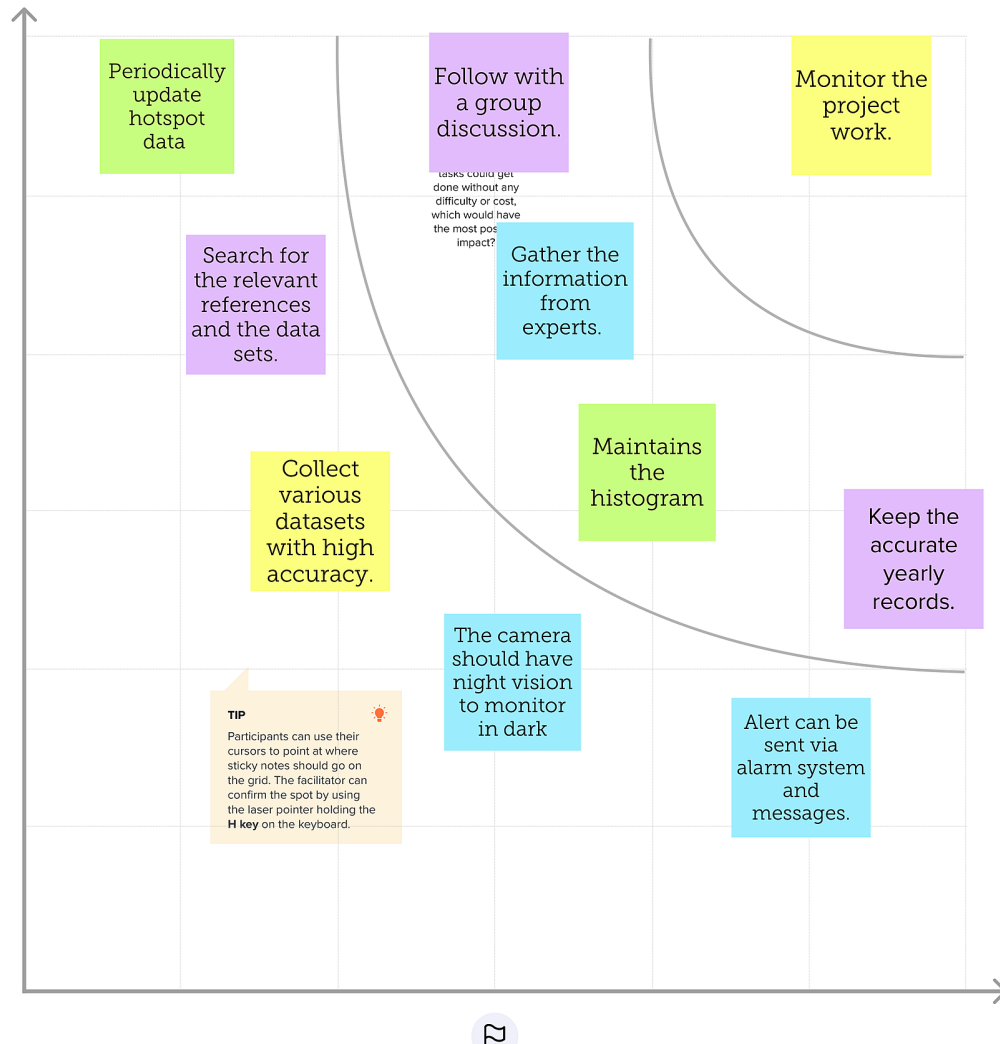
Maha Priya.T



Group ideas:



Prioritize:



3.3.PROPOSED SOLUTION:

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>1.Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives.</p> <p>2.It is difficult to predict and detect Forest Fire in a sparsely populated forest area.</p>

		3. So, it is necessary to detect the fire in an early stage to control it.
2.	Idea / Solution description	1. The model will detect forest fires automatically with the help of image processing in deep learning, with the use of images and videos given by the user to observe, detect and report fire events.
3.	Novelty / Uniqueness	When the fire is detected, the station will get a notification via message and an alarm system will be activated automatically to alert the user.
4.	Social Impact / Customer Satisfaction	1. This can reduce the forest fire in the beginning stage, by alerting users. 2. The user can also use this as a surveillance 3. Camera to monitor the forest. Saving the most essential Forest cover.
5.	Business Model (Revenue Model)	1. This application will be available in a subscription-based model. 2. Supply chain, power & supply, Fire stations, and government by providing services.
6	Scalability of the Solution	1. This application can monitor different places simultaneously and can

		<p>detect fire accurately</p> <p>2.This application can handle a large number of users and data simultaneously.</p> <p>number of users and data simultaneously.</p>
--	--	---

3.4.PROPOSED SOLUTION FIT:

Project Title:Early Detection of forest fire using deep learning

Project Design Phase-I Solution Fit Template

Team ID: PNT2022TMID49620

Define CS, fit into CC	1.CUSTOMER SERVICE CS <p>1.Tribal people and forest department officers living in forest.</p> <p>2.Animals , birds and other living things in the forest.</p>	6. CUSTOMER CONSTRAINTS CC <p>1.Solar power cameras can be used as a power source</p> <p>2.Waterproof cameras.</p> <p>3.Seamless connection.</p>	5. AVAILABLE SOLUTIONS AS <p>1.Notification is sent via messages.</p> <p>2.Fire alarm is activated to nearby stations.</p>	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P <p>1.Detecting small fire sparks in forest becomes difficult.</p> <p>2.Camera should always be in motion</p>	9. PROBLEM ROOT CAUSE RC <p>1.Special analysis system can be used.</p> <p>2. Wireless mobile network via SIM can be used transfer alert message throughout areas.</p>	7. BEHAVIOUR BE <p>1.Climate change should be monitored.</p> <p>2.Hot areas should be monitored clearly.</p>	
Identify strong TR & EM	3. TRIGGERS TR <p>1.Correct detection.</p> <p>2.Alarm alert</p> <p>3.Follow correct algorithm</p>	10. YOUR SOLUTION SI <p>1.Mobile application can be developed for specific areas.</p> <p>2.Forest can be monitored by several cameras.</p> <p>3.This can be used in wild life sanctuaries.</p>	8.CHANNELS of BEHAVIOUR CH <p>ONLINE Connected directly to the user via Internet.</p> <p>OFFLINE Alerts can be sent via Offline messages and an alarm system is activated.</p>	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM <p>BEFORE</p> <p>1.Unable to detect small sparks.</p> <p>2.camera should always be in motion.</p> <p>AFTER</p> <p>1.Able to detect small sparks.</p> <p>2. 360 view of camera is used.</p>			

4.REQUIREMENT ANALYSIS

4.1.FUNCTIONAL REQUIREMENTS:

FR No	Functional Requirement(Epic)	SubRequirement(Story /Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Image recognition	The system shall be able to take real inputs of satellites images and determine whether image contains fire or not.
FR-4	Forest Monitoring	Forest are monitored 24/7 through
FR-5	Alert	The system will send notification to the user when fire is detected
FR-6	Detection	The system shall take training sets of fire and checks for fire or no fire or smoke
FR-7	Operating system	The system can run as a service on Windows .

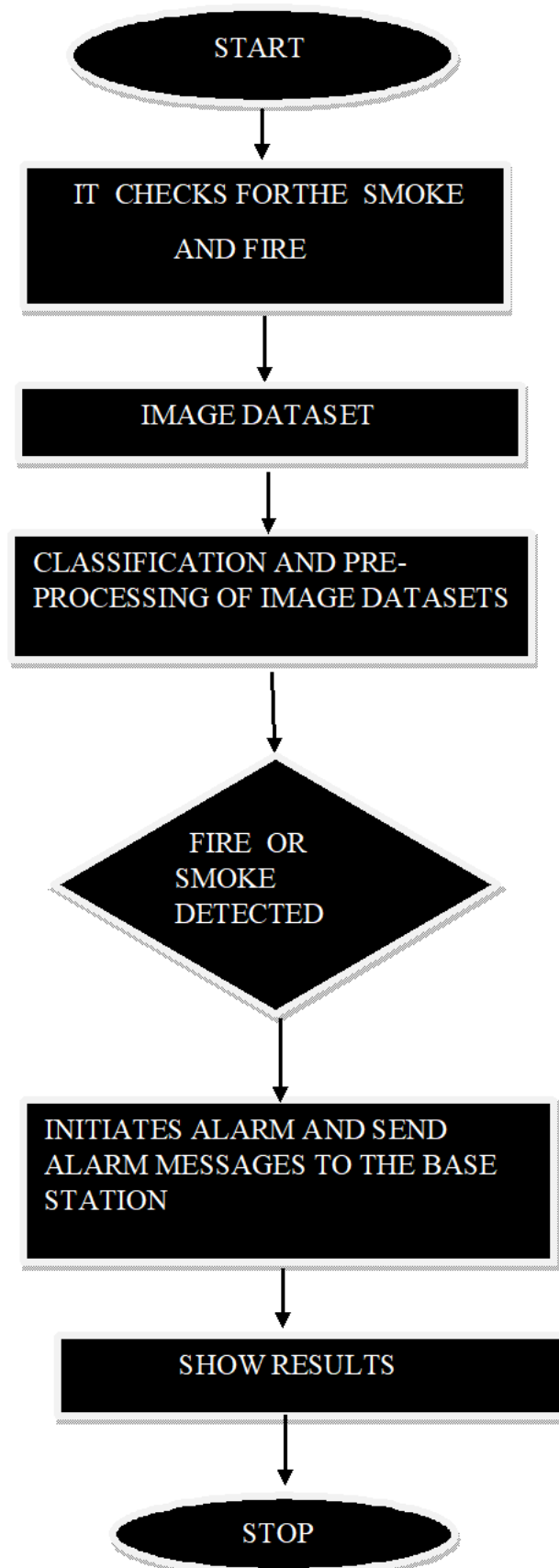
4.2. NON-FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description

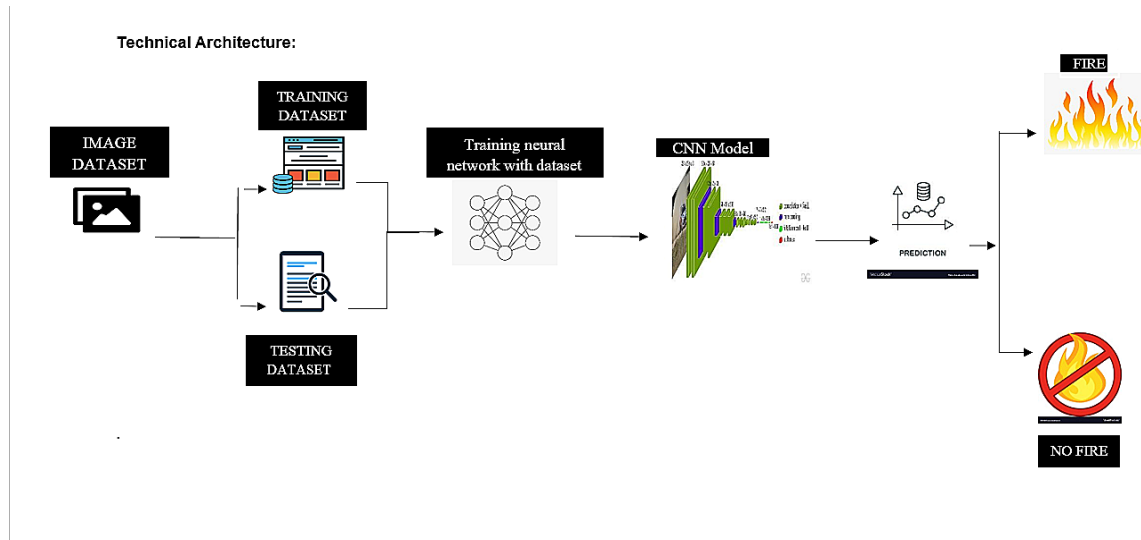
NFR-1	Usability	Model is user friendly to use and very effective.
NFR-2	Security	More secure environment.
NFR-3	Reliability	Model is safe to install.
NFR-4	Performance	Model will achieve high accuracy.
NFR-5	Availability	Build model is available in all the time
NFR-6	Scalability	Model can handle large amount of data and can easily adapt to every environment.
NFR-7	Testability	Putting in more training data into the model can improve the accuracy level of the system.

5.PROJECT DESIGN

5.1.DATA FLOW DIAGRAMS:



5.2.SOLUTION AND TECHNICAL ARCHIETECTURE:



5.3.USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	UserStory/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user,I can register for the application by entering my email, password, and confirming my password.	I can access my account/dashb oa rd	High	Sprint-1
		USN-2	As a user, I	I can receive	High	Sprint-1

			will receive confirmation email on cell phone have registered for the application	confirmation email & click confirm		
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user , I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user,I can log into the application by entering email & password		High	Sprint-1
	Dashboard					

6.PROJECT PLANNING & SCHEDULING

6.1.SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Download data set	USN-1	The data is downloaded from the Kaggle website and then the data set is classified into training and testing images.	10	High	U. Sakthi malavika T.Mahapriya R.Subasri S.Yoga pandeeswari
Sprint-1	Image preprocessing	USN-1	In Image processing technique the first step is usually importing the libraries that will be needed in the program. Import Keras library from that library and import the ImageDataGenerator Library to the Python script. The next step is defining the arguments for the ImageDataGenerator . Here the arguments which we are given inside the image data generator class are, rescale,	10	High	U. Sakthi malavika T.Mahapriya R.Subasri S.Yoga pandeeswari

			shear_range, rotation range of image, and zoom range that we can consider for images. The next step is applying the ImageDataGenerat or 10 High S. Kamalakar S. Shaamini C. Selshia Mary R. Maharaj arguments to the train and test dataset.			
Sprint-2	Training image	USN-2	In this training phase the ImageDataGenerato rargu ments is applied to the training images and the model is tested with several images and the model is saved.	20	High	U. Sakthi malavika T.Mahapriya R.Subasri S.Yoga pandeeswari
Sprint-3	Testing image	USN-3	In this testing phase the Image processing techniques is applied to the testing images and executed for prediction	20	High	U. Sakthi malavika T.Mahapriya R.Subasri S.Yoga pandeeswari
Sprint-4	Evaluati on metrics and accuracy	USN-4	In this phase the result, prediction, accuracy, and performance of the	20	High	U. Sakthi malavika T.Mahapriya R.Subasri S.Yoga pandeeswari

			model are tested.			
--	--	--	-------------------	--	--	--

MILESTONE & ACTIVITY LIST:

Activity Number	Activity Name	Detailed Activity Description	Task Assigned	Status
1.1	Access Resources	Access the resources (courses) in project dashboard.	All Members	COMPLETED
1.2	Rocket chat registration	Join the mentoring channel via platform & rocket-chat mobile app.	All Members	COMPLETED
1.3	Access workspace	Access the guided project workspace.	All Members	COMPLETED
1.4	IBM Cloud registration	Register on IBM Academic Initiative & Apply Feature code for IBM Cloud Credits.	All Members	COMPLETED
1.5	Project Repository Creation	Create GitHub account & collaborate with Project Repository in project workspace.	All Members	COMPLETED
1.6	Environment Setup	Set-up the Laptop / Computers based on the pre-requisites for each technology track.	All Members	COMPLETED
2.1	Literature survey	Literature survey on the selected project &	All Members	COMPLETED

		Information Gathering.		
2.2	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
2.3	Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problems statements	All Members	COMPLETED
2.4	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
2.5	Brainstorming	List the ideas (at least 4 per each team member) by organizing the brainstorm session and prioritize the ideas	All Members	COMPLETED
2.6	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
3.1	Proposed Solution Document	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	All Members	COMPLETED
3.2	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED

3.3	Problem - Solution fit & Solution Architecture	Prepare problem - solution fit document & Solution Architecture.	All Members	COMPLETED
3.4	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
4.1	Customer Journey Map	Prepare Milestone & Activity List.	All Members	COMPLETED
4.2	Technology Training	Prepare Sprint Delivery Plan.	All Members	COMPLETED
4.3	Functional Requirements & Data Flow Diagrams	Collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc.	All Members	COMPLETED
4.4	Technology Architecture	Importing the ImageDataGenerator Library	All Members	COMPLETED
4.5	Technology Training	Define the parameters/arguments for ImageDataGenerator class.	All Members	COMPLETED
5.1	Milestone & Activity List	Applying ImageDataGenerator functionality to trainset and test set.	All Members	COMPLETED
5.2	Sprint Delivery Plan	Importing the model building libraries.	All Members	COMPLETED
6	Data Collection	Initializing the model.	All Members	COMPLETED
7.1	Image	Adding CNN Layers.	All Members	COMPLETED

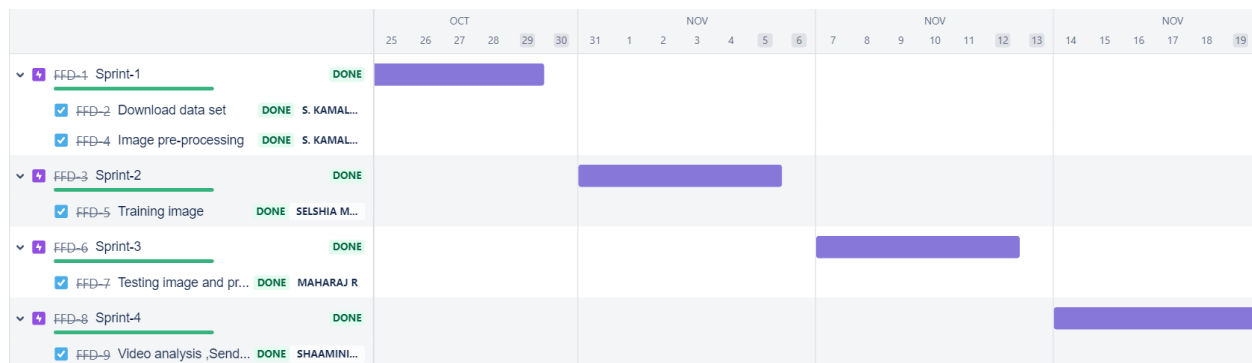
	Preprocessing			
7.2	Image Preprocessing	Adding Dense Layers	All Members	COMPLETED
7.3	Image Preprocessing	Configuring the learning process	All Members	COMPLETED
8.1	Model Building	Training the Model	All Members	COMPLETED
8.2	Model Building	Save the model	All Members	COMPLETED
8.3	Model Building	Predictions	All Members	COMPLETED
8.4	Video Analysis	OpenCV for video processing	All Members	COMPLETED
8.5	Video Analysis	Creating an account in Twilio service.	All Members	COMPLETED
8.6	Video Analysis	Sending alert message.	All Members	COMPLETED
8.7	Train CNN Model on IBM	Register for IBM Cloud	All Members	COMPLETED

6.2.SPRINT DELIVERY SCHEDULE:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct	05 Nov	20	05 Nov

			2022	2022		2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3.REPORTS FROM JIRA:



7.CODING & SOLUTIONING

7.1.FEATURE 1:

1.IMAGE DATA GENERATOR:

Keras ImageDataGenerator is used for getting the input of the original data and further, it makes the transformation of this data on a random basis and gives the output resultant containing only the data that is newly transformed. It does not add the data. from keras.preprocessing.image import ImageDataGenerator

2.PARAMETERS

2.1.Rescale:

The ImageDataGenerator class can be used to rescale pixel values from the range of 0-255 to the range 0-1 preferred for neural network models. Scaling data to the range of 0-1 is traditionally referred to as normalization.

2.2.Shear Range:

Shear range means that the image will be distorted along an axis, mostly to create or rectify the perception angles. It's usually used to augment images so that computers can see how humans see things from different angles.

2.3.Rotation range:

ImageDataGenerator class allows you to randomly rotate images through any degree between 0 and 360 by providing an integer value in the rotation_range argument. When the image is rotated, some pixels will move outside the image and leave an empty area that needs to be filled in.

2.4.Zoom Range:

The zoom augmentation method is used to zooming the image. This method randomly zooms the image either by zooming in or it adds some pixels around the image to enlarge the image. This method uses the zoom_range argument of the ImageDataGenerator class. It can specify the percentage value of the zooms either in a float, range in the form of an array.

2.5.Horizontal Flip:

Horizontal flip basically flips both rows and columns horizontally. So for this, It have to pass the horizontal_flip=True argument in the ImageDataGenerator constructor.

3.CONVOLUTION NEURAL NETWORK:

A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice. The layers used in the CNN is Convolutional ,maxpooling, and flatten layer.

3.1.Convolutional Layer:

A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image Convolution layer is used for a image processing to blur and sharpen images, but also to perform other operations. from keras.layers import Convolution2D

3.2.Maxpooling Layer:

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. `from keras.layers import MaxPooling2D`

3.3.Flatten Layer:

Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image. `from keras.layers import Flatten`

4.DENSE LAYER:

Dense Layer is used to classify image based on output from convolutional layers.

7.2.FEATURE 2(CODE):

Importing Keras libraries

```
import keras
```

Importing ImageDataGenerator

```
from Keras from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
```

Defining the Parameters

```
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,shear_range=0.2,rotation_range=
180,zoom_range=0.2,horizontal_flip=True)
```

Applying ImageDataGenerator functionality to train dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM
PROJECT/dataset/DATASET/archive/Dataset/Dataset/train_set',target_size=(64,64),batch_size=32,class_mode='binary')
```

Applying ImageDataGenerator functionality to test dataset

```
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IBM
```

PROJECT/dataset/DATA

SET/archive/Dataset/Dataset/test_set',target_size=(64,64),batch_size=32,class_mode='binary')

Importing Model Building Libraries

#to define the linear Initialisation import sequential

from keras.models import Sequential

#to add layers import Dense

from keras.layers import Dense

#to create Convolutional 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')#to define the linear Initialisation import sequential

from keras.models import Sequential #to add layers import Dense from keras.layers

import Dense #to create Convolutional 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()

Adding CNN Layers

model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))

#add maxpooling layers

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

#add faltten layer

model.add(Flatten())

Add Dense layers

#add hidden layers

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

#add output layer

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

configuring the learning process

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

Training the model

```
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4,validation_freq=4)
```

Save the model

```
model.save("forest.h5")
```

Predictions

```
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
import numpy as np
#import cv2
import cv2
#load the saved model
model=load_model("/content/drive/MyDrive/IBM PROJECT/dataset/forest.h5")
img=image.load_img('/content/drive/MyDrive/IBM PROJECT/dataset/DATASET/archive/Dataset/Dataset/test_set/with fire/FORESTFIRE (1).jpg')
x=image.img_to_array(img)
res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
pred=model.predict(x)
pred = int(pred[0][0])
pred
int(pred)
pip install twilio
from twilio.rest import Client
if pred==0:
    print('Forest fire')
    account_sid='AC0f20fb7b8e71118fa14d874dc2384676'
```

```

auth_token='74902c8f190f5a4d288bbf5e3b48c84e'
client=Client(account_sid,auth_token)
message=client.messages \
.create(
body='forest fire is detected,stay alert',
#use twilio free number
from_='+18608542959',
#to number
to='+916380889559')
print(message.sid)
print("Fire detected")
print("SMS Sent!")
elif pred==1:
print('No Fire')

```

Open cv for video processing

```

pip install twilio
from logging import WARNING
#import opencv library
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
#import playsound package

```

Creating An Account in Twilio Service

Sending Alert Message

```

import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
import librosa

```

```

from tensorflow.keras.preprocessing import image
from keras.models import load_model
# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
cap = cv2.VideoCapture('/content/drive/MyDrive/IBM
PROJECT/dataset/datasetvideo.mp4')
# Check if camera opened successfully
if (cap.isOpened() == False):
    print("Error opening video stream or file")
# Read until video is completed
while(cap.isOpened()):
    # Capture frame-by-frame
    ret, frame = cap.read()
    if ret == True:

        cv2.imshow('frame', frame)
        x = image.img_to_array(frame)
        res = cv2.resize(x, dsize=(64, 64), interpolation=cv2.INTER_CUBIC)
        # expand the image shape
        x = np.expand_dims(res, axis=0)
        model = load_model("/content/drive/MyDrive/IBM PROJECT/dataset/forest.h5")
        pred = model.predict(x)
        pred = int(pred[0][0])
        pred
        int(pred)
        if pred == 0:
            print('Forest fire')
            break
        else:
            print("no danger")
            break
# When everything done, release the video capture object
cap.release()
# Closes all the frames
cv2.destroyAllWindows()
from twilio.rest import Client
if pred == 0:

```

```

print('Forest fire')
from twilio.rest import Client
account_sid='AC0f20fb7b8e71118fa14d874dc2384676'
auth_token='74902c8f190f5a4d288bbf5e3b48c84e'
client=Client(account_sid,auth_token)
message=client.messages \
.create(
body='forest fire is detected,stay alert',
#use twilio free number
from_='+18608542959',
#to number
to='+916380889559')
print(message.sid)
print("Fire detected")
print("SMS Sent!")
elif pred==1:
print('No Fire')

```

8.TESTING

8.1.Test Cases:

8.2.User Acceptance Testing:

Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of the [Early detection of forest fire using Deep Learning] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severit y1	Severit y2	Severit y3	Severit y4	Sub total

By Design	5	1	1	1	8
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	7	2	4	10	23
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won'tFix	0	3	2	1	6
Totals	15	9	11	14	49

Test Case Analysis:

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Out source Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Final Report Output	2	0	0	2

9.RESULTS

9.1.PERFORMANCSE METRICS:

S.No.	Parameter	Values
-------	-----------	--------

1	Model Summary	As a threat of forest fire increases due to climate changes, the need for finding a detection system increases .The proposed Deep Learning-based model to predict early detection of forest fire. The Proposed model successfully classifies the images into fire and no fire, and sends an alert messages in case of fire. Thus, the Deep Learning algorithms proved their efficiency in detecting different objects
2	Accuracy	Training Accuracy - 98% Validation Accuracy - 95%

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- 1.Ability to cover areas at different altitudes and locations.
- 2.The results is quite accurate with the accuracy upto 95% .
- 3.Reliability - The model is very effective, inexpensive and easy to apply.
- 4.The model, it shows the 'fire' and 'no fire' images classified with high accuracy.
- 5.Video analysis of this model leads to low degree of misjudgment of fire detection.

DISADVANTAGES:

- 1.Individual learner is responsible for learning global information to avoid false positives.
- 2.The limited learning and perception ability of individual learners is not sufficient to make them perform well in complex tasks.
- 3.Proper connectivity and maintenance will be a complex task.

11.CONCLUSION

As a threat of forest fire increases due to climate changes, the need for finding a detection system increase .The proposed Deep Learning-based model to predict the early detection of forest fire. The Proposed model successfully classifies the images into fire and no fire, and sends an alert messages in case of fire. Thus, the Deep Learning algorithm proved their efficiency in detecting the forest fire.

12.FUTURE SCOPE

- Integrate live satellite data and process real time processing of the fires.
- Enhance the time complexity of the detection of forest fires to improve the speed.
- These accidents can be controlled to a greater extent.
- Forest fire leads to destruction of excess of species, by using this technique it will save the life and environment.

13.APPENDIX

SOURCE CODE:

Source code-

<https://colab.research.google.com/drive/1XDoscqoV6cljSkqQxnhoLoy33aYayrWq>

Our Github link - <https://github.com/IBM-EPBL/IBM-Project-31290-1660198661>

DEMO VIDEO:

Demo video link - <https://www.youtube.com/embed/MxI4361OxGE>