EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

A PROJECT REPORT

Submitted by

PANDI MEENA R **953219106023**
LAVANYA L **953219106018**
PONCHENDILA P **953219106024**
SUBASHINI P **953219106038**

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

UNIVERSITY VOC COLLEGE OF ENGINEERING

THOOTHUKUDI-628 008

ANNA UNIVERSITY: CHENNAI 600025

NOV 2022

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this Project report **"EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRE"** is the bonafide work of

PANDI MEENA R	**953219106023**
LAVANYA L	**953219106018**
PONCHENDILA P	**953219106024**
SUBASHINI P	**953219106038**

who carried out the project under my supervision

SIGNATURE

Dr. K. ESAKKI MUTHU

HEAD OF THE DEPARTMENT

Department of Electronics and

Communication Engineering,

University VOC College of Engineering,

Thoothukudi-628008

SIGNATURE

Dr. A. MOOKAMBIGA

MENTOR

Department of Electronics and

Communication Engineering,

University VOC College of Engineering,

Thoothukudi-628008

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**	
	13.1	Source Code
	13.2	GitHub & Project Demo Link

ABSTRACT

As a new fire detection technology, image fire detection has recently played a crucial role in reducing fire losses by alarming users early through early fire detection. Image fire detection is based on an algorithmic analysis of images. However, there is a lower accuracy, delayed detection, and a large amount of computation in common detection algorithms, including manually and machine automatically extracting image features. Therefore, novel image fire detection algorithms based on the advanced object detection CNN models. A comparison of the proposed and current algorithms reveals that the accuracy of fire detection algorithms based on object detection CNNs is higher than other algorithms. A comparison of the proposed and current algorithms reveals that the accuracy of fire detection algorithms based on object detection CNNs is higher than other algorithms. With rapid economic development, the increasing scale and complexity of constructions has introduced great challenges in fire control. Therefore, early fire detection and alarm with high sensitivity and accuracy is essential to reduce fire losses. However, traditional fire detection technologies, like smoke and heat detectors, are not suitable for large spaces, complex buildings, or spaces with many disturbances. Due to the limitations of above detection technologies, missed detections, false alarms, detection delays and other problems often occur, making it even more difficult to achieve early fire warnings. Recently, image fire detection has many advantages such as early fire detection, high accuracy, flexible system installation, and the capability to effectively detect fires in large spaces. It processes image data from a camera by algorithms to determine the presence of a fire or fire risk in images. Therefore, the detection algorithm is the core of this technology, directly determining the performance of the image fire detector. There are three main stages in the process of image fire detection algorithms, including image pre-processing, feature extraction, and fire detection.

INTRODUCTION

PROJECT OVERVIEW

In this project, 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 sensors installed in the buildings. They are slow and cost inefficient due to their primitive design and technology. Our Project critically analyses the scope of Artificial intelligence for detection and sending alert messages. This project uses self-built dataset. The data is then pre-processed and use the CNN to build a Deep learning model. The test set of the dataset is given as input for validating the algorithm and experiments are noted. The project focus on building a highly accurate model that can be used in almost any use case of fire detection. The forest images in given dataset is pre-processed and the model is trained using the given dataset. Using IBM Watson Studio Service the image classification model can be saved in our local system and that model can be used for video analysis for fire detection.

PURPOSE

The main objective of this project is to accurately detect the fire and help to confine fire to limited areas before much damage occurs. This system will be very effective in preventing occurrance of false alert messages. The model uses Twillio service to send the alert message when fire is detected. The technologies used are python libraries like keras, Tensorflow, Image Data Generator, open CV and IBM Watson studio service and Twillio service.

LITERATURE SURVEY

2.1 Existing problem

54.40% of forests in India are exposed to occasional fires,7.49% to moderate fires and 2.40% to severe fires. Fire detection system increases the response time as they alert the correct people to extinguish the fire in initial stage hence minimizing the destruction caused by forest fire.

Nearly 85% of wild land fires in the United States are caused by humans. Human-caused fires result from campfires left unattended, the burning of debris, equipment use and malfunctions, negligently discarded cigarettes and the intentional acts of arson. Lightning is one of the natural causes of fires. So fire detection system is essential to avoid such destructive event.

2.2 AVAILABLE SOUTIONS

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.IEEE REGION10 Conference(TENCON),2020, pp. 11361140,doi:10.1109/tencon 50793.2020.9293722.

1. The system involves pre-processing the image data and applying data augmentation such as shearing, flipping, etc.
2. It uses models like VGG16 , ResNet50 , and DenseNet121 for the classification of images.

3. 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.

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

S.NO	AUTHOR	TITLE	NAME OF JOURNAL
2.	Byron Arteaga, Mauricio Diaz, Mario jajoa, University of Naino Pasto Columbia .	Deep Learning Applied forest Fire Detection.	2020 IEEE International Symposium on signal processing and information Technology(ISSPIT),2020,pp, 16,doi:10,1109/ISSPIT51521.2020.9408859.

1. The data processing was done through open source programming language Python, the cloud service Googlecollab, and deep learning algorithms using Pytorch's library.
2. 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.
3. The classification of images is done by using the pre-trained models of ResNet and VGG pre-trained models.
4. To validate the performance of each pre-trained model the k-fold method is used.
5. The model obtained during the validationis sent to Raspberry to test its functionality.

S.NO	AUTHOR	TITLE	NAME OF JOURNAL
3.	Raghad k. Mohammed(Department 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.

1. The proposed framework aims to detect smoke and fire based on the images received from the video stream from the Raspberry Pi
2. Pre-processing of image data.
3. Image data augmentation (Scale, horizontal flip, and vertical flip).
4. Pre-training model imagenet dataset ->{inception-ResNet-V2}.
5. By fine-tuning the above two steps we have to send that to the fully connected layer with softmax.
6. 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.	Fire Detection Using Deep Learning.	International Journal of Progressive Research in Science And Engineering Volume-1,Issue-5,August-2020.

1. The model is divided into two parts
2. a. Data collection and Pre-processing.
3. b. Building fire detection model by transfer learning.
4. 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.
5. The second step is to extract the video features of pre-trained models using Keras.
6. 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.

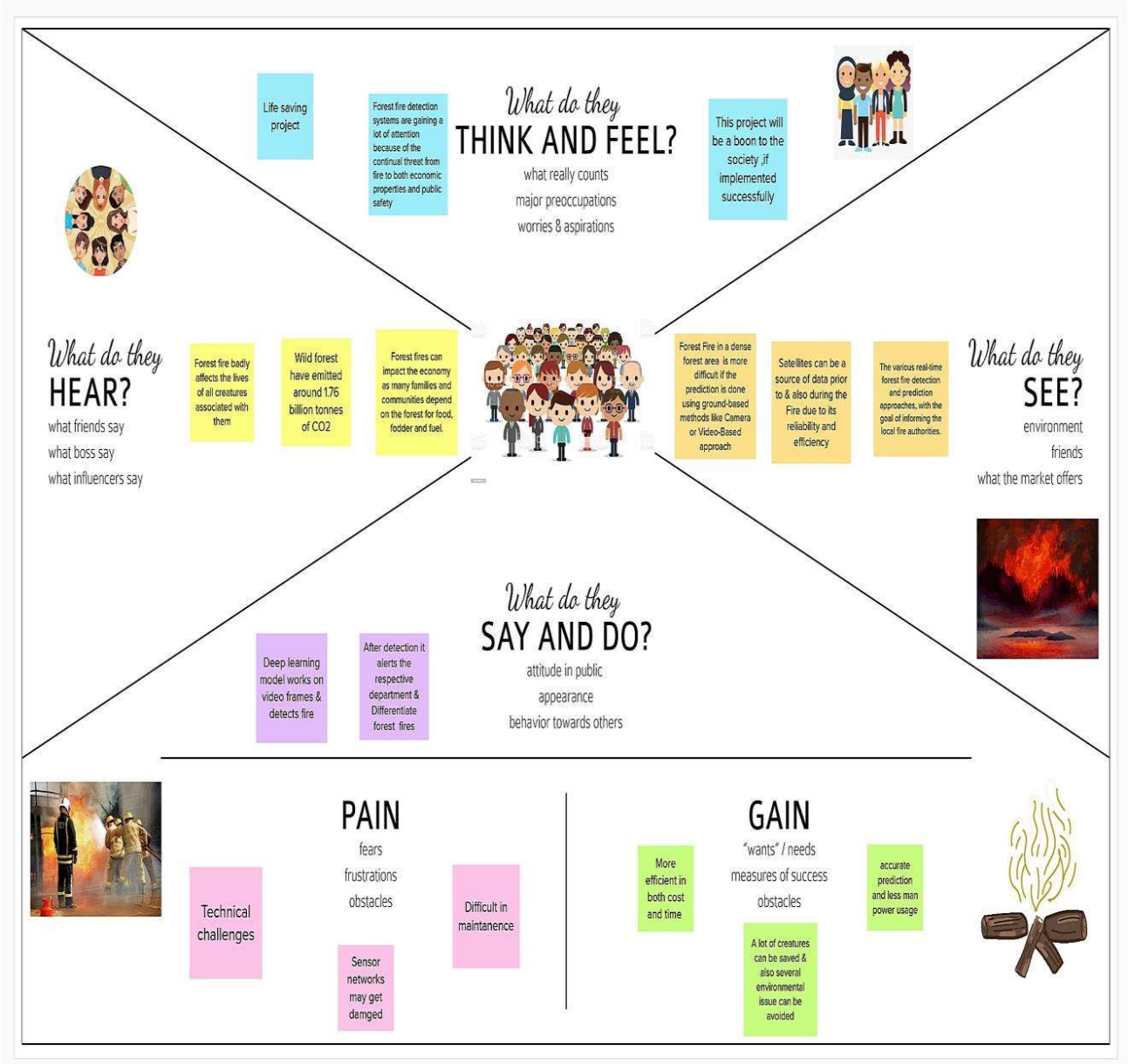
2.3 PROBLEM STATEMENT DEFINITION

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.

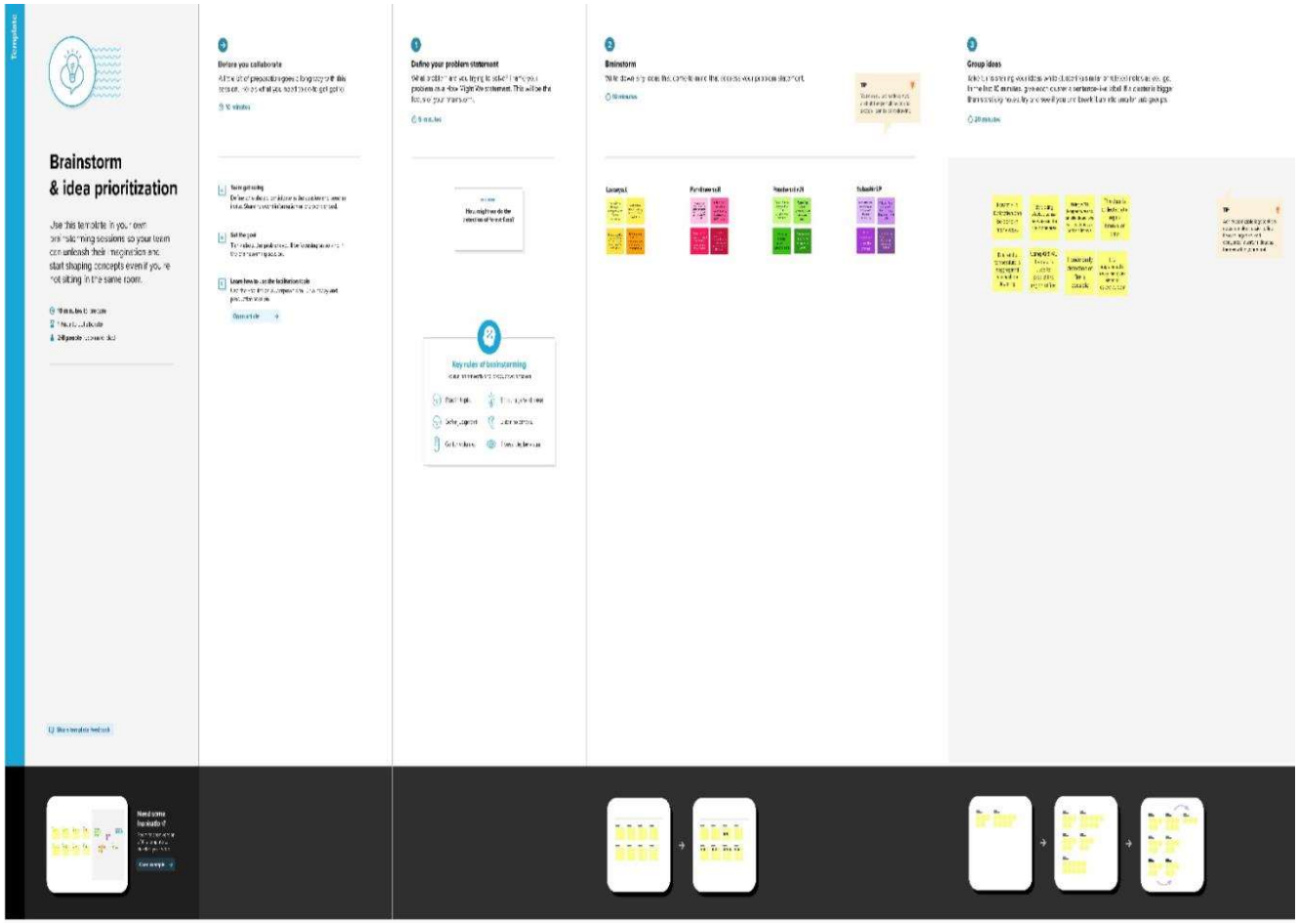
Who does the problem affect?	People living in the forest.
When does the issue occurs?	When there is a climate change in the environment .
Where is the issue occurring?	The issue occurs when there is a difficulty to identify the forest fires.
What is the issue?	Forest fires are a major environmental issue,creating 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.

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



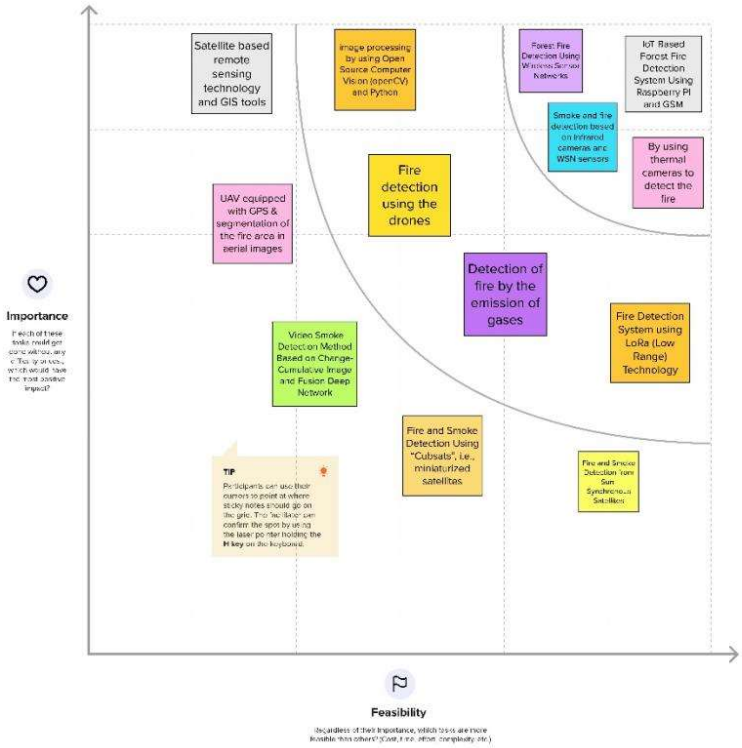
3.2 IDEATION & BRAINSTORMING



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
•	Problem Statement (Problem to be solved)	Forest fires are one of the most worrisome natural disasters ,destroying thousands of acres of forest and nearby urban zones, affecting plants, animals and human life.
•	Idea / Solution description	Use Deep Learning model to detect fire in forest area which could accurate predictions at the ignition instances
•	Novelty / Uniqueness	CNN model deployed using python libraries have a more positive outlook, and they detect forest fire in earliest before it spread to larger area.
•	Social Impact / Customer Satisfaction	Wildfires can disrupt transportation, communications, powe rand water supply. They also lead to a deterioration of the air quality, loss of property, crops, resources, animals and people.
•	Business Model (Revenue Model)	The proposed method will be implemented using the Artificial Intelligence. It creates more intelligent detection algorithms for the deployed model
•	Scalability of the Solution	Deployed CNN model is trained with a dataset accordingly to detect fire and send an alert message to concerned user .

3.4 PROBLEM SOLUTION FIT

PROJECT DESIGN PHASE-1 PROBLEM SOLUTION FIT

Project Title: Emerging methods for Early Detection of Forest Fire

Team ID: PNT2022TMID50802

Define CS, fit into CC

1 . CUSTOMER SEGMENT CS

→ Federal agencies (forest fire management) such as National Disaster Management. Authority(NDMA) USDA's Forest Service
→ The Department of the Interior's Affairs, Bureau of Land Management, Fish and wildlife Service and National Park Service.
→ People living around fire prone forest area.

6. CUSTOMER CONSTRAINTS CC

Major constraints are that there should a stable and uninterrupted network availability for the wireless sensor network to operate, High cost and power consumption of sensor nodes.

5. AVAILABLE SOLUTONS AS

Wireless Sensor nodes are deployed to detect fire in starting stage that cover entire forest area. UAV equipped with GPS are employed to localize the fire area in aerial images. In early days, Smoke alarms and watch towers are used .**Pros:** Lives could be saved . **Cons:** Network availability , high power consumption.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE/ PROBLEMS J&P

54.40% of forests in India are exposed to occasional fires,7.49% to moderate fires and 2.40% to severe fires. Fire detection system increases the response time as they alert the correct people to extinguish the fire in initial stage hence minimizing the destruction caused by forest fire.

9. PROBLEM ROOT CAUSE RC

Nearly 85% of wild land fires in the United States are caused by humans. Human-caused fires result from campfires left unattended, the burning of debris, equipment use and malfunctions, negligently discarded cigarettes and the intentional acts of arson. Lightning is one of the natural causes of fires. So fire detection system like wireless sensor nodes are essential to avoid such destructive event.

7. BEHAVIOUR BE

The fire detection is done on the basis of ARMSTRONG FIRE INDEX along with the values of gas sensors. In case of forest fire breaks out, a message to the concerned authority is sent first and then data collected will be uploaded in a database from the base station computer to an online website.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Natural cause: High temperature causes combustion of dry leaves, Lightning etc,.
Manmade: Unattended campfires, arson.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE: Loss of biodiversity, Extinction of plants and animals, Loss of valuable timber.
AFTER: Flora and Fauna could be saved efficiently using the forest fire surveillance system.

10. YOUR SOLUTION SL

Fire detection using Wireless Sensor Network technologies normally deploy a large number of sensors densely that can observe fire prone area by gathering information, transform it into electrical signals, send it to a remote location to do analysis and rescue operations. Hence there is no need to build towers or complicated communication links. It can be deployed even in inaccessible places providing a real time information on forest fire at ignition instance.

8. CHANNELS OF BEHAVIOR CH

8.1 ONLINE: Collect the data and form a Dataset to compare the Temperatures of flame regions for forest fire detection.

8.2 OFFLINE: Having a proper spark arrestors, shovels , fire extinguishers and store a reservoir with water or sand.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

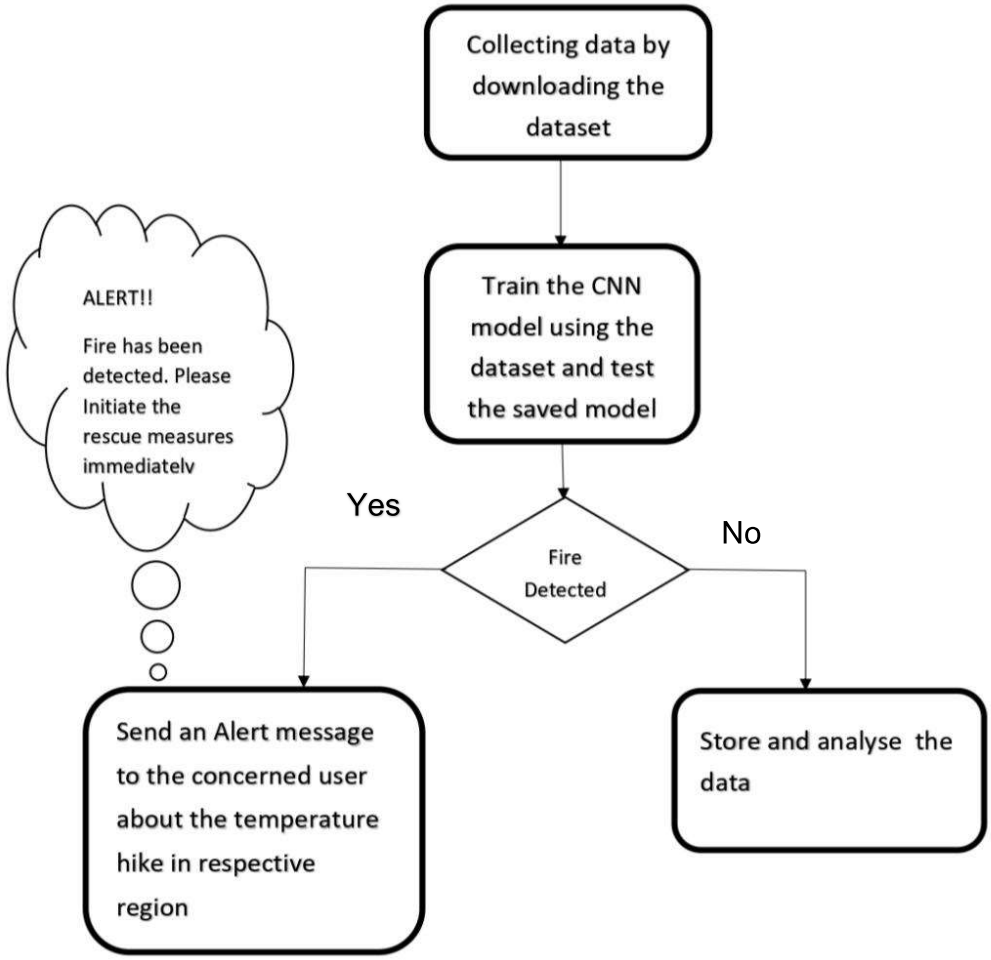
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration in Twillio service with a valid mobile number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Login credentials	Login using a number and password given during the software installation also user can change the password by clicking on new password and also it have an option offorget password.
FR-4	User interface	The system will provide a graphical user interface(GUI).The user can able to view all required data regarding fire alert signal.
FR-5	Communication Interface	On detecting fire alert message is sent to the user.

4.2 NON-FUNCTIONAL REQUIREMENTS

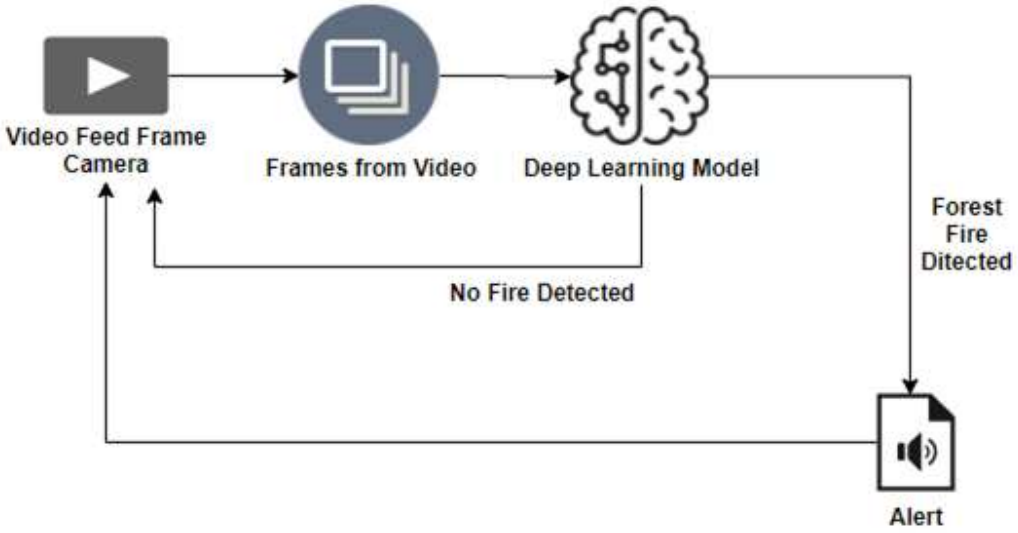
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Our product is very user friendly and the installation of hardware &software interface is also very easy
NFR-2	**Security**	Using Twillio servvice the alert message is send to concerned user more securely . The Trained model has high scale of security since it can be accessed in local system by extracting the model using IBM Watson Studio
NFR-3	**Reliability**	The model is highly reliable ,it could predict the fire accurately as it was trained by the required dataset in prior.
NFR-4	**Performance**	When the model predict the fire with in a fraction of seconds alert message will be sent to the concerned user using Twillio service
NFR-5	**Availability**	The user need not to be available all the time for monitoring.Whenever the model predict the fire ,it automatically sends the alert message to client without internet
NFR-6	**Scalability**	Our system is scalable enough to sense many throughput at the same time and software can support many no. of visit at same time while maintaining optimal performance

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

UserType	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/dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email on cell phone have	I can receive confirmation email	High	Sprint-1

			registered for the application	& 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

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	Data Collection	USN-1	Download the dataset	20	High	R.Pandimeena L.Lavanya N.Ponchendila P.Subashini
Sprint-1	Image Preprocessing	USN-2	Importing the image data generator and defining the arguments and applying them to Train set and Testset	20	Low	R.Pandimeena L.Lavanya N.Ponchendila P.Subashini
Sprint-2	Model Building	USN-3	Import model building libraries, adding CNN Layers and Training the model	20	Medium	R.Pandimeena L.Lavanya N.Ponchendila P.Subashini
Sprint-3	Video Analysis	USN-4	Video is Processed using open CV, after installing Twilio Service model id	10	High	R.Pandimeena L.Lavanya N.Ponchendila P.Subashini

			saved using Keraslibra ry			
Sprint-4	Train CNN model on IBM	USN-5	Train and save the model on IBM , Download the modelto Local systemand Test it.	20	High	R.Pandimeena L.Lavanya N.Ponchendila P.Subashini

MILESTONE AND ACTIVITYLIST

S.No	MILESTONE	DESCRIPTION	DURATION	WORKING STATUS
1	Project Objectives	Project objectives are what you plan to achieve by the end of your project. This might include deliverables and assets, or more intangible objectives like increasing productivity or motivation.	1 WEEK	COMPLETED
2	Data Collection	Artificial Intelligence is a data hunger technology, it depends heavily on data, without data, it is	1 WEEK	COMPLETED

		impossible for a machine to learn. It is the most crucial aspect that makesalgorithm training possible.		
3	Image Processing a. Keras	Image Pre-processing includesthe following main tasks • Import ImageDataGenera torLibrary. • Configure ImageDataGenera torClass. • Applying ImageDataGenera torfunctionality to the trainset and test set.	1 WEEK	COMPLETED
4	Model Building	The neural network model is to be built by adding different network layers like convolution,pooling, flattening, dropout and neural layers. In this milestone, we startbuilding our model by: 1.Initializing the model 2.Adding Convolution layers	1 WEEK	COMPLETED

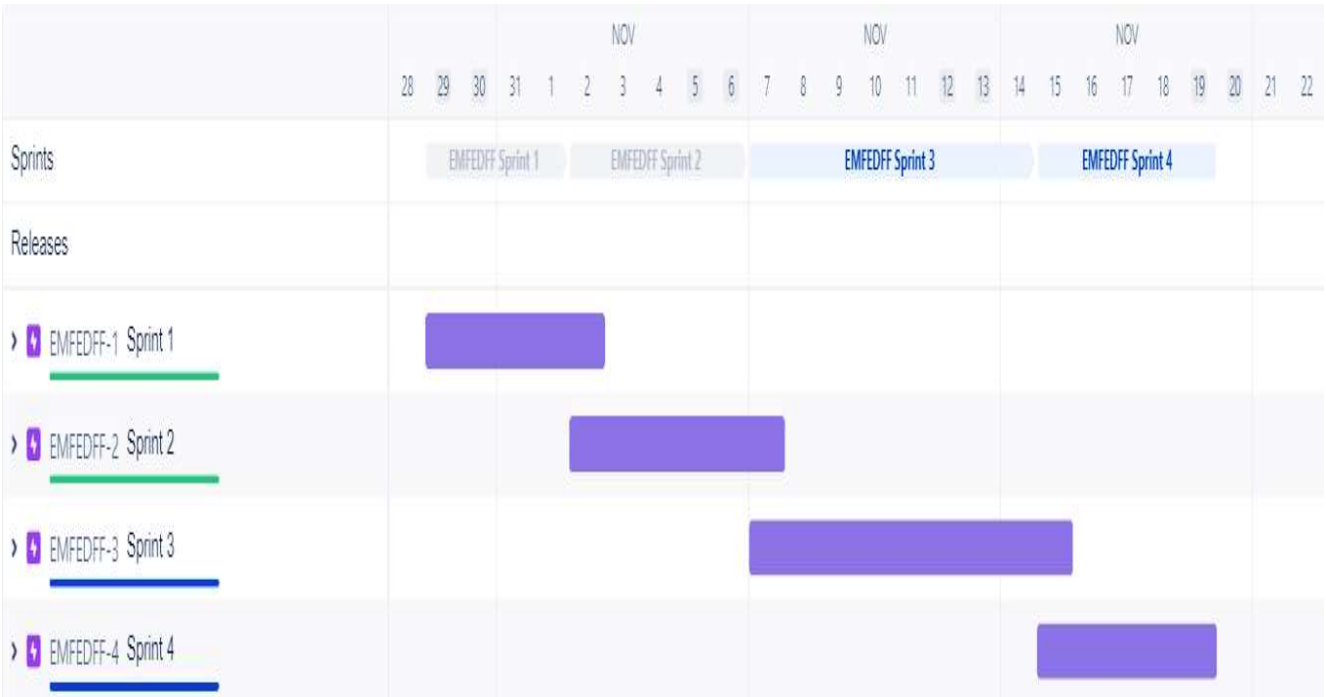
		3. Adding Pooling layers4.Flatten layer 5.Full connection layers whichinclude hiddenlayers		
5	Video Analysis	In this process we open CV forImage processing, Creating anaccount in Twilio Service, Sendingalert messages.	2 WEEKS	COMPLETED
6	Train CNN Model on IBM	Train the Image classification Models on IBM Cloud using IBMWatson Studio Service.	2WEEKS	COMPLETED
7	Ideation Phase	Ideation is the process where we generate Literature survey on the selected project & gathered information, Empathy map, ideation.	1 WEEK	COMPLETED
8	Project Design Phase I	Project design phase I is the process where we design Proposedsolution, Problem solution fit, Solution architecture. The aim is todevelop one or more designs that can be used to achieve the desired project goals.	1 WEEK	COMPLETED

9	Project Design Phase II	Project design phase II is the process where we design Customer journey, Functional requirement, Data flow diagram, Technology architecture.	1 WEEK	COMPLETED
10	Project Planning Phase	Project planning phase is the process where we prepare Milestone & activity list, Sprint delivery plan.	1 WEEK	COMPLETED
11	Project Development Phase	Project development phase is the process where we develop Sprint 1, Sprint 2, Sprint3, Sprint 4.	2 WEEKS	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	4 Days	29 Oct 2022	02 Nov2022	20	09 Nov 2022
Sprint-2	20	5 Days	02 Nov 2022	07 Nov 2022	20	12 Nov 2022
Sprint-3	20	8 Days	07 Nov 2022	15 Nov 2022	20	15 Nov 2022
Sprint-4	20	9 Days	15 Nov 2022	19Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1

1.IMAGE DATA GENERATOR:

Keras Image Data Generator 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. pre-processing.image import Image Data  Generator
```

2.PARAMETERS

2.1.Rescale:

The Image Data Generator 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:

Image Data Generator 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 Image Data Generator 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 Image Data Generator 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)

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

IMAGE PREPROCESSING

```
#importing keras library
import keras
```

```
#importing the image data generator
from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
```

```
#Defining the parameter for image generator class
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_ra
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
#Applying image data generator functionality to train set
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/train_set',
target_size=(128,128),batch_size=32,class_mode='binary')
```

```
#Applying image data generator functionality to test set
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/test_set',
target_size=(128, 128),batch_size=32,class_mode='binary')
```

MODEL BUILDING

```
#To define linear intialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To creat Convolution kernal 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 convolution 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 convolution layer
model.add(Convolution2D(64,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(128,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(128,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add flatten layer
model.add(Flatten())

model.add(Dense(512,activation='relu'))
model.add(Dense(1,activation='sigmoid'))

#configuring the learning process
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])


#Training the model
r=model.fit(x_train,epochs=10,validation_data=x_test)


#save the model
model.save("forestalert.h5")
```

VIDEO ANALYSIS

```
pip install twilio


from twilio.rest import Client


#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/forestalert.h5")
img=image.load_img('/content/drive/MyDrive/Dataset/test_set/with fire/Uttarakhand_forest_f
x=image.img_to_array(img)
# res=cv2.resize(x,dsize=(150,150),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(x,axis=0)


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

import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
import librosa
import tensorflow
from tensorflow.keras.preprocessing import image
from keras.models import load_model
from google.colab import drive
from google.colab.patches import cv2_imshow



# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
video = cv2.VideoCapture(r'/content/drive/MyDrive/Wild fire.mp4')
name=['forest','with fire']


while(1):
 success,frame = video.read()
 cv2.imwrite("image.jpg",frame)
 img = tensorflow.keras.utils.load_img("image.jpg",target_size = (128,128))
 x = image.img_to_array(img)
 x = np.expand_dims(x,axis = 0)
 pred = model.predict(x)
 pred = pred[0][0]
 if pred > 0.5:
   pred = 1
 else :
    pred = 0
 print(pred)
 cv2.putText(frame,"predicted class = "+str(name[pred]),(100,100),cv2.FONT_HERSHEY_SIMPLEX
 if pred==1:
  account_sid = 'ACab5b7ac22466b88a9cda7cf5414b750a'
  auth_token = 'c9c95130eade17e5e3d3f936283bef7a'
  client = Client(account_sid, auth_token)
  message = client.messages \
    .create(
    body='Forest Fire is detected,Stay alert',
    from_='+17088477470',
    to='+918825826199')
  print(message.sid)
```

```
    print("Fire detected")
    print("SMS Sent!")
    cv2_imshow(frame)
    break
   else:
    print("No Danger")
    break
  cv2_imshow(frame)
  if cv2.waitKey(1) & 0xFF == ord('a'):
   break
  video.release()
 cv2.destroyAllWindows()
```

8. TESTING

8.1 Test Cases

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.r
```

IMAGE PREPROCESSING

```
#importing keras library
import keras
```

```
#importing the image data generator
from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
```

```
#Defining the parameter for image generator class
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_ra
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
#Applying image data generator functionality to train set
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/train_set',
target_size=(128,128),batch_size=32,class_mode='binary')
```

```
Found 436 images belonging to 2 classes.
```

```
#Applying image data generator functionality to test set
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/test_set',
target_size=(128, 128),batch_size=32,class_mode='binary')
```

```
Found 121 images belonging to 2 classes.
```

MODEL BUILDING

```
#To define linear intialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To creat Convolution kernal 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 convolution 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 convolution layer
model.add(Convolution2D(64,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(128,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(128,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add flatten layer
model.add(Flatten())

model.add(Dense(512,activation='relu'))
model.add(Dense(1,activation='sigmoid'))


#configuring the learning process
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])


#Training the model
r=model.fit(x_train,epochs=10,validation_data=x_test)
```

```
Epoch 1/10
14/14 [==============================] - 234s 16s/step - loss: 0.6142 - accuracy: 0.(
Epoch 2/10
14/14 [==============================] - 17s 1s/step - loss: 0.3169 - accuracy: 0.86
Epoch 3/10
14/14 [==============================] - 19s 1s/step - loss: 0.4177 - accuracy: 0.87
Epoch 4/10
14/14 [==============================] - 17s 1s/step - loss: 0.3900 - accuracy: 0.86
Epoch 5/10
14/14 [==============================] - 17s 1s/step - loss: 0.2594 - accuracy: 0.89
Epoch 6/10
14/14 [==============================] - 19s 1s/step - loss: 0.1766 - accuracy: 0.92
Epoch 7/10
14/14 [==============================] - 17s 1s/step - loss: 0.1475 - accuracy: 0.94
Epoch 8/10
14/14 [==============================] - 19s 1s/step - loss: 0.1356 - accuracy: 0.94
Epoch 9/10
14/14 [==============================] - 17s 1s/step - loss: 0.1492 - accuracy: 0.93
Epoch 10/10
14/14 [==============================] - 19s 1s/step - loss: 0.1470 - accuracy: 0.94
```

```
#save the model
model.save("forestalert.h5")
```

VIDEO ANALYSIS

```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting twilio
  Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 17.6 MB/s
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packa
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3
```

```
from twilio.rest import Client


#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/forestalert.h5")
img=image.load_img('/content/drive/MyDrive/Dataset/test_set/with fire/Uttarakhand_forest_f
x=image.img_to_array(img)
# res=cv2.resize(x,dsize=(150,150),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(x,axis=0)


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


import cv2
```

```
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
import librosa
import tensorflow
from tensorflow.keras.preprocessing import image
from keras.models import load_model
from google.colab import drive
from google.colab.patches import cv2_imshow
```

WITH FIRE

```
# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
video = cv2.VideoCapture(r'/content/drive/MyDrive/Wild fire.mp4')
name=['forest','with fire']


while(1):
 success,frame = video.read()
 cv2.imwrite("image.jpg",frame)
 img = tensorflow.keras.utils.load_img("image.jpg",target_size = (128,128))
 x = image.img_to_array(img)
 x = np.expand_dims(x,axis = 0)
 pred = model.predict(x)
 pred = pred[0][0]
 if pred > 0.5:
   pred = 1
 else :
    pred = 0
 print(pred)
 cv2.putText(frame,"predicted class = "+str(name[pred]),(100,100),cv2.FONT_HERSHEY_SIMPLEX
 if pred==1:
  account_sid = 'ACab5b7ac22466b88a9cda7cf5414b750a'
  auth_token = 'c9c95130eade17e5e3d3f936283bef7a'
  client = Client(account_sid, auth_token)
  message = client.messages \
    .create(
    body='Forest Fire is detected,Stay alert',
    from_='+[redacted]',
    to='+91[redacted]')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
  cv2_imshow(frame)
  break
 else:
  print("No Danger")
  break
 cv2_imshow(frame)
 if cv2.waitKey(1) & 0xFF == ord('a'):
```

```
    break
video.release()
cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 15ms/step
1
SM5f311301d5eb273b1cc7685535fde506
Fire detected
SMS Sent!
```

WITHOUT FIRE

```
# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
video = cv2.VideoCapture(r'/content/drive/MyDrive/Forest.mp4')


while(1):
 success,frame = video.read()
 cv2.imwrite("image.jpg",frame)
 img = tensorflow.keras.utils.load_img("image.jpg",target_size = (128,128))
 x = image.img_to_array(img)
 x = np.expand_dims(x,axis = 0)
 pred = model.predict(x)
 pred = pred[0][0]
 if pred > 0.5:
   pred = 1
 else :
    pred = 0
 print(pred)
 cv2.putText(frame,"predicted class = "+str(name[pred]),(100,100),cv2.FONT_HERSHEY_SIMPLEX
 if pred==1:
  account_sid = 'ACab5b7ac22466b88a9cda7cf5414b750a'
  auth_token = 'c9c95130eade17e5e3d3f936283bef7a'
  client = Client(account_sid, auth_token)
  message = client.messages \
    .create(
    body='Forest Fire is detected,Stay alert',
    from_='+1[redacted],
    to='+91[redacted]')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
  cv2_imshow(frame)
  break
 else:
  print("No Danger")
  cv2_imshow(frame)
  break
 if cv2.waitKey(1) & 0xFF == ord('a'):
  break
 video.release()
cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 15ms/step
0
No Danger
```

9. RESULTS

9.1.PERFORMANCE 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 up to 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 mis-judgment of fire detection.
6.CNN model automatically detects the important features without human supervision.

DISADVANTAGES:

1.CNN has several layers, so training process takes lots of time if the computer does not has good GPU .
2.Overfitting, exploding gradient and class imbalance are the major challenges while training the model using CNN.
3.Proper connectivity and maintenance will be a complex task.
4.Lots of Training Data required.

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.