MEENAKSHI COLLEGE OF ENGINEERING

B.TECH-INFORMATION TECHNOLOGY

INTERNET OF THINGS

LITERATURE SURVEY

EMERGING METHOD FOR EARLY DETECTION OF FOREST FIRE

TEAM ID PNT2022TMID27778

TEAM MEMBERS:

TEAM LEADER- M.PRIYANKA

M.RAHAT SAFANA

P.S.SNEHA

S.SWETHA

1.INTRODUCTION

1.1 PROJECT OVERVIEW

This has a negative impact on the ecology, especially when thereis a forest fire, and makes it highly hazardous for animals to staythere. An automated system that can identify any fire situation through any of the alarm systems is needed to prevent such losses. This study examines the development, applications, and momentum of the Internet of Things in the field of firefighting.The article also survey that was carried out to determine research trends and challenges in fire initiatives. The fire Internet of Things(IoT) connects numerous items with organizations within the fire domain as its major objective. Unmanned aerial vehicle (UAV) is a new type of aircraft that has been utilized in the existing system the disadvantages of thispaper is based on only smoke image observed from the camera. Sometimes it not detect correctly. So in proposed system we have used color format called YCbCR is a family of color space used to clarify the clear image of fire during forest fire .

1.2 PURPOSE

Forest fires as of late have been annihilating both for normal biological system, biodiversity and woodland economy. With expanding populace weight and change in worldwide atmosphere situation, there is an expansion in level of fires that are a significant reason for declining Indian woodlands. As indicated by woodland study report of India, 50 % of backwoods regions in nation are fire inclined (going from 50 to 90 % in certain conditions of nation). Around 6 % of the woods are inclined to extreme fire harms. The reason for this planned framework is tomanufacture a dependable fire location framework so as to know dynamic status ofbackwoods temperature in specific conditions. It is about the sensors and dynamic checking framework to dodge a significant fire and genuine harm to woods.

1. LITERATURE SURVEY

2.1 EXISTING PROBLEM

The fire Internet of Things(IoT) connects numerous items with organizations within the fire domain as its major objective. Unmanned aerial vehicle (UAV) is a new type of aircraft that has been utilized in the existing system the disadvantages of this paper is based on only smoke image observed from the camera. Sometimes it not detect correctly.

2.2REFERENCE

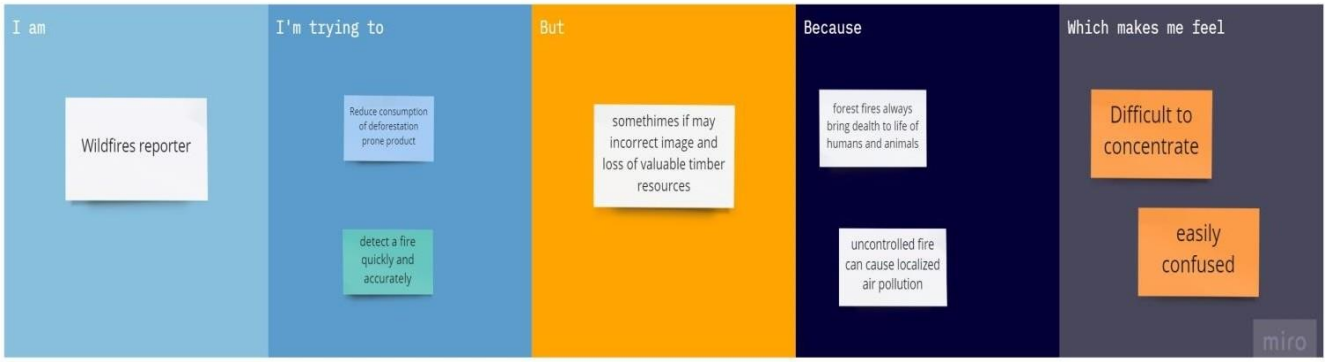
1.Yuan C, Zhang Y M, Liu Z X. A survey on technologies for automatic forest fire monitori fighting using unmanned aerial vehicles and remote sensing techniques[J]. Canadian Journal ofForest Research, 2015.

2.Piccinini P, Calderara S, Cucchiara R. Reliable smoke detection in the domains of image energy and color[C]. The 15thIEEE International Conference on Image Processing, 2008 3.Tung T X, Kim J M. An effective four-stage smoke-detection algorithm using video images for early fire-alarm systems[J]. Fire Safety Journal, 2011.

3.Toreyin B U, Dedeoglu Y, Cetin A E. Contour based smoke detection in video using wavelets[C]. 2006 European Signal Processing Conference.2006

4.Cucchiara R, Grana C, Piccardi M. Detecting moving objects, ghosts, and shadows in video streams[J]. IEEE Transactions on Pattern Analysis Machine Intelligence, 2003.

2.3 PROBLEM STATEMENT

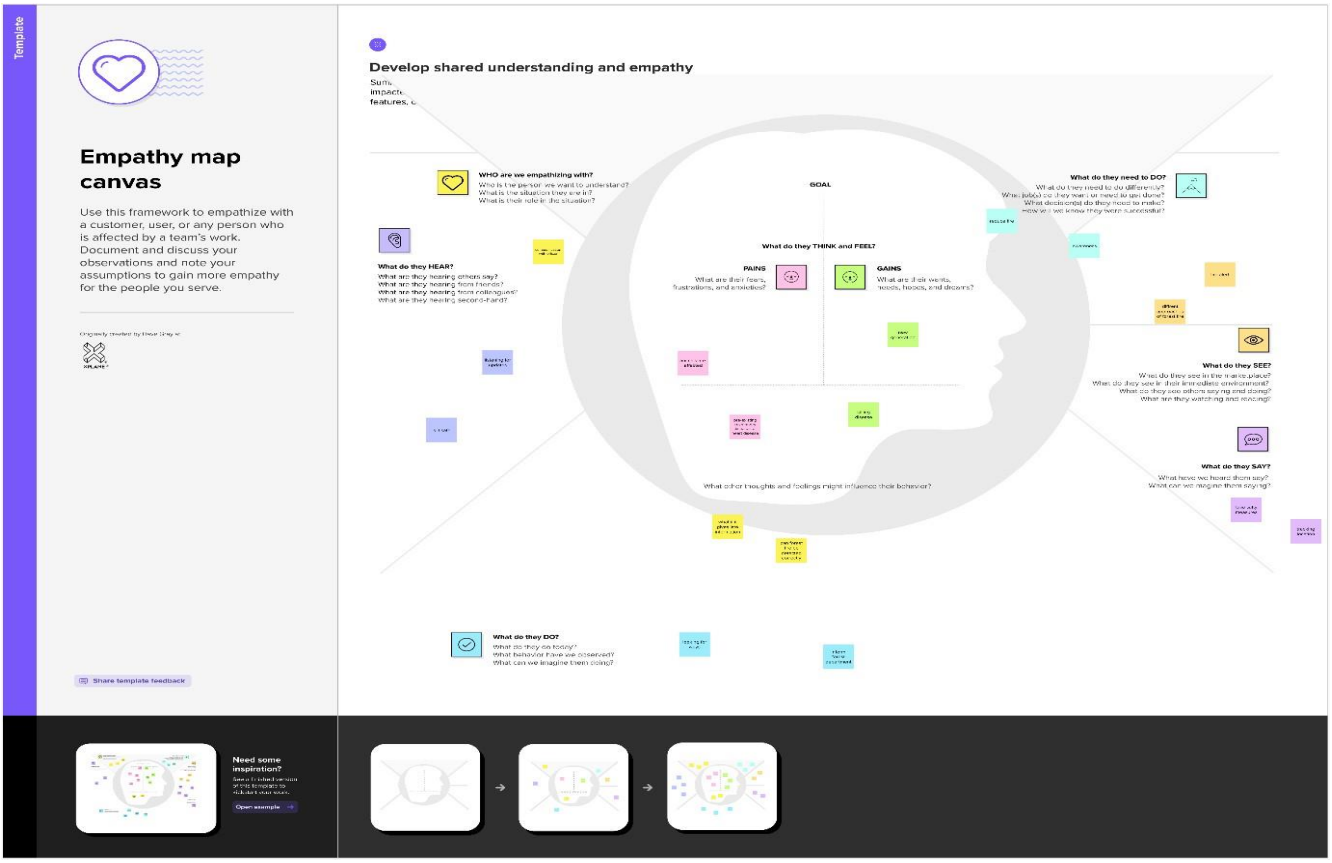


Problem Statement (PS)	Iam (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Wildfires reporter	Reduce consumption of deforestation prone product	Sometime it may incorrect image and loss of valuable timber resources	Forest fires bring death to life of humans and animals	Difficult to concentrate
PS-2		Detect a fire quickly and accurately		Uncontrolled fires cause localized air pollution	Easily confused

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas:

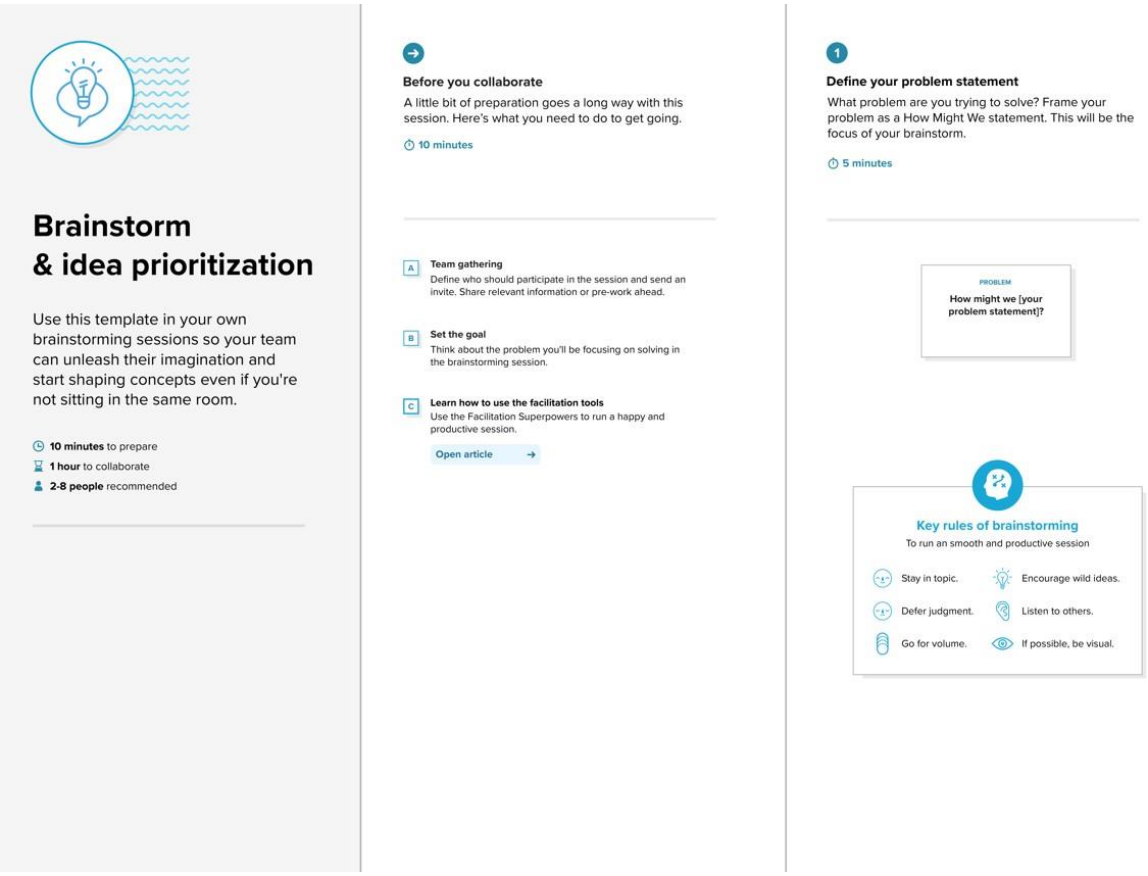
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



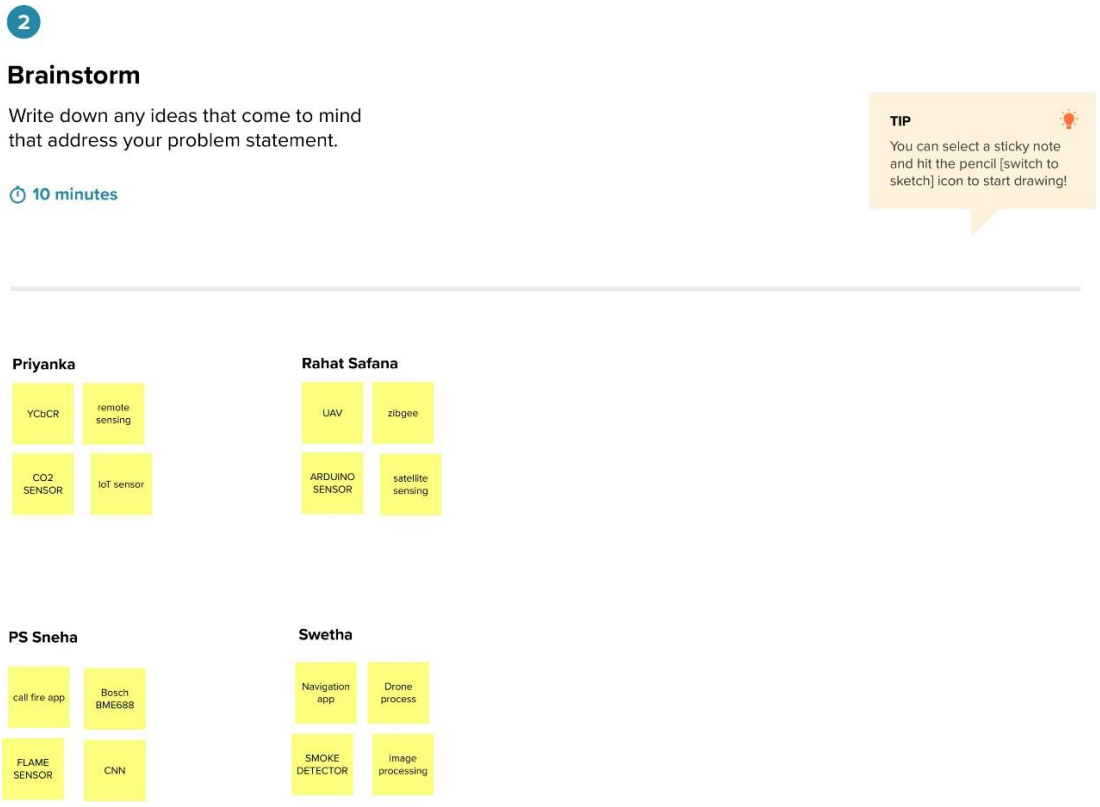
3.2 IDEATION AND BRAINSTORM

Brainstorming provides a free and open environment that encourages everyone within a teamto participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.Use this template in your own brainstorming sessions so your team can unleash theirimagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



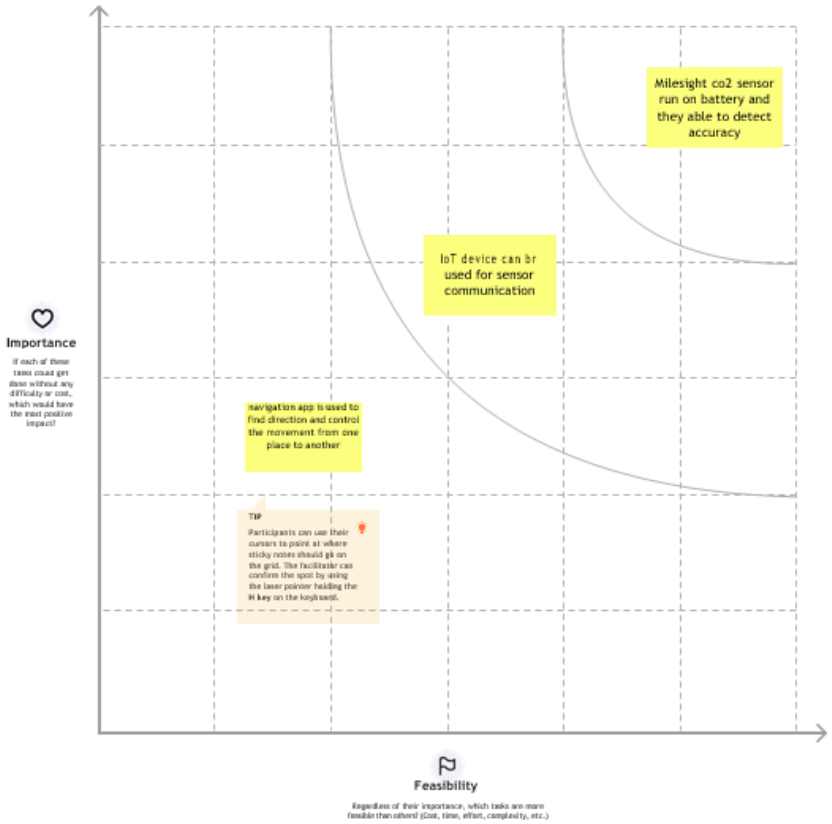
Step-3: Idea Prioritization

4

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.

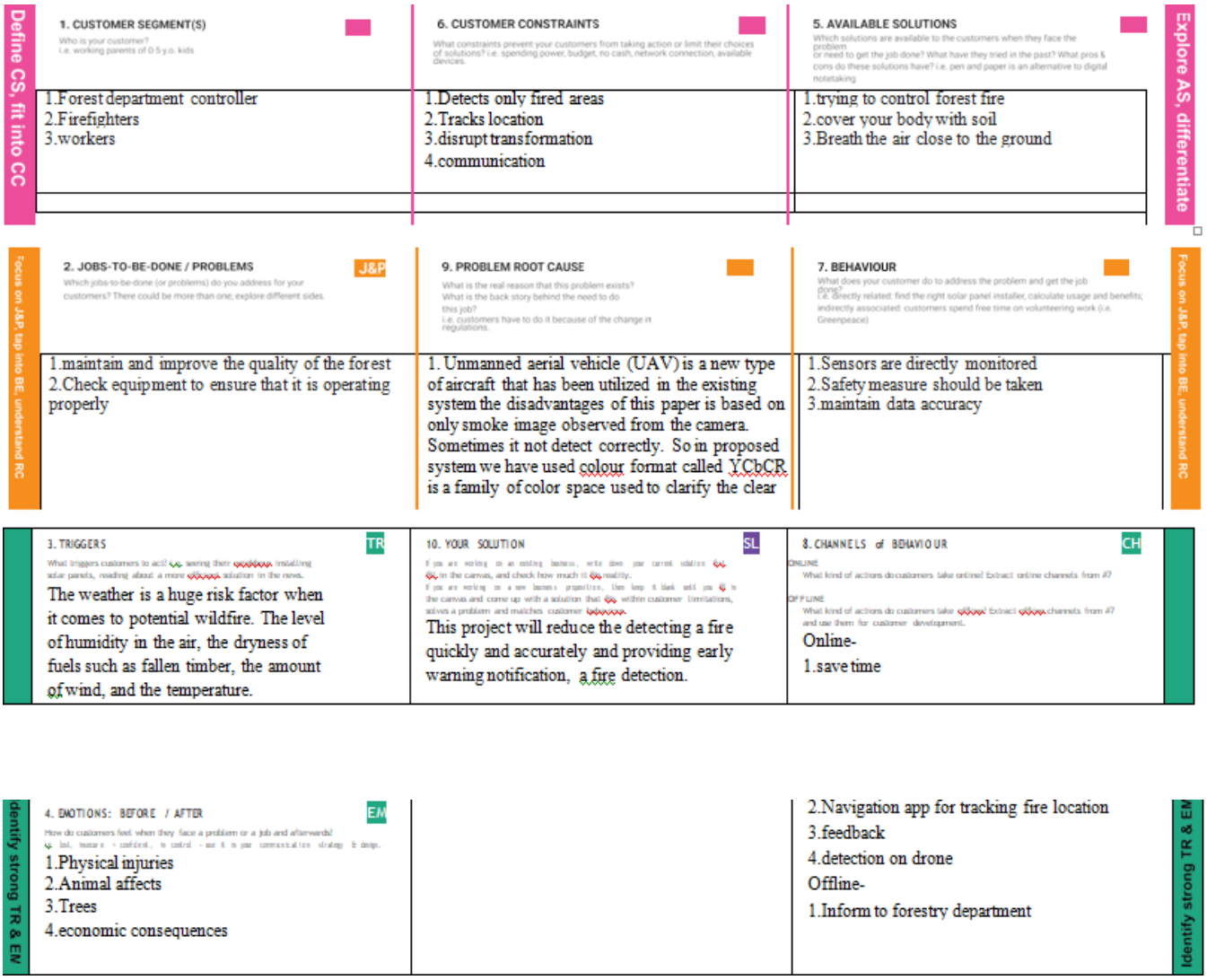
20 minutes



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	This project will reduce the detecting a fire quickly and accurately and providing early warning notification, a fire detection
2.	Idea / Solution description	Provides colour based YCbCR to give better image.
3.	Novelty / Uniqueness	Detect of forest fire using bosch sensor
4.	Social Impact / Customer Satisfaction	Cause air pollution, threaten biodiversity and spoil the aesthetics of an area
5.	Business Model (Revenue Model)	Sings with forest fire detection
6.	Scalability of the Solution	Using the bosch and navigation every wildfire detection is less than 60 sec

3.3 PROPOSED SOLUTION FIT



4. REQUIRMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Getting alert message from sensor
FR-2	User Confirmation	Conform the location by navigation app

4.2 Non-functional Requirements:

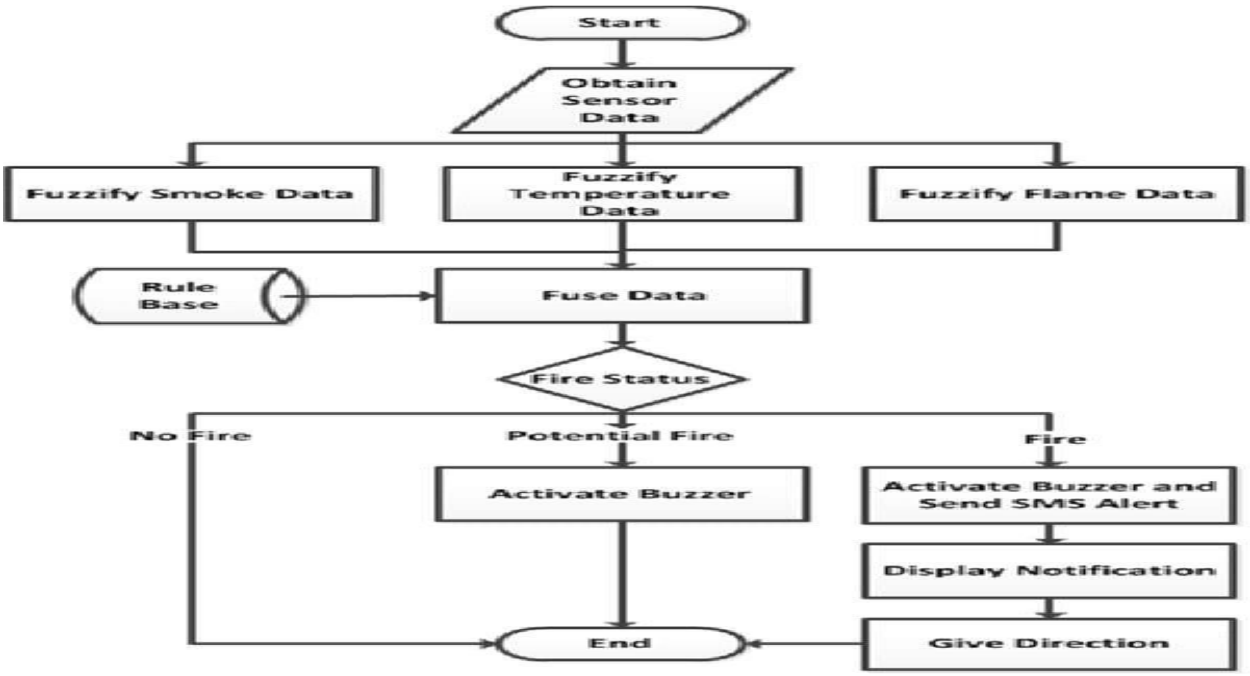
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Temperature sensor and smoke sensor are deployedat certain distances so that the whole forest area can be kept inside the view in order to detect the ignition alarming temperature and the range of carbon dioxide gas (CO2).
NFR-2	**Security**	Inserting image processing to the sensor are help to detect fire
NFR-3	**Reliability**	It reduces fires and awareness and tracks fired location correctly
NFR-4	**Performance**	When fires gets detected in sensor it identify the fire and alert sounds within 5 sec
NFR-5	**Availability**	It will detect the fire quickly and accurately 24/7, security.
NFR-6	**Scalability**	It can identify the all captured fire at the same time

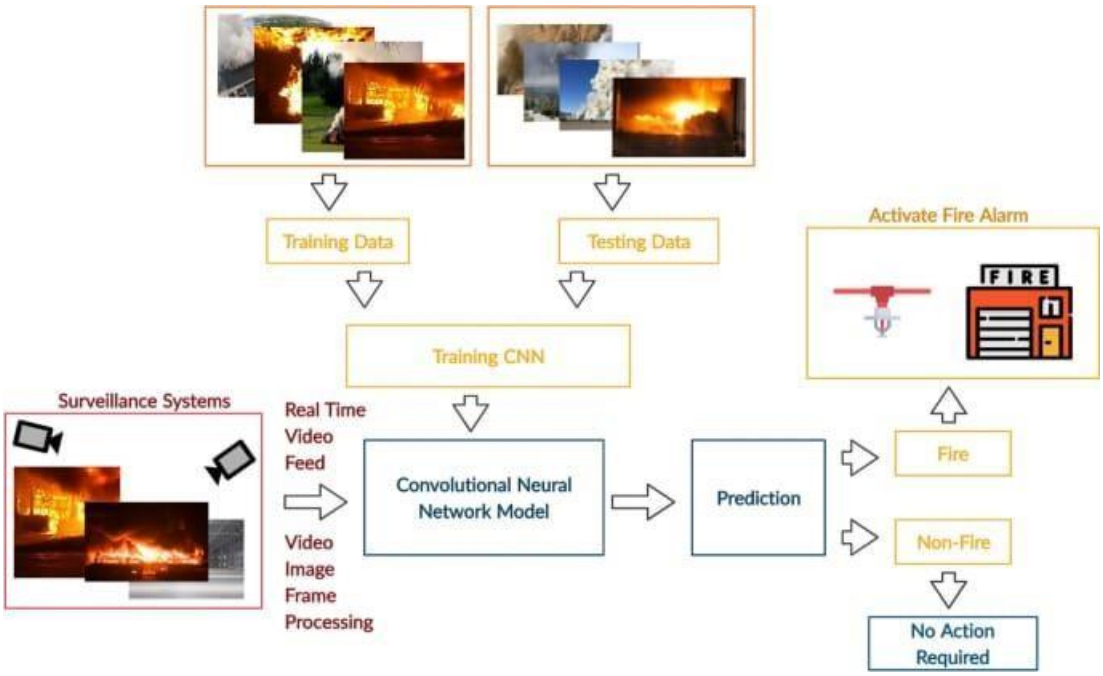
5.PROJECT DESIGN

5.1 Data Flow Diagrams:

The traditional visual representation of how information moves through a system is a data flow diagram (DFD). Atidy and understandable DFD can graphically represent the appropriate amount of the system requirement. It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story I Task	Acceptance criteria	Priority	Release
Environmentalist	Collect the data	USN-1	As an Environmentalist.it is necessary to collect the data of the forest which includes temperature,humidity,wind and rain of the forest	It is necessary to collect the right data else the prediction may become wrona	High	Sprint-1
		USN-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithms	Medium	Sprint-2
	Implement Algorithm	USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	High	Sprint-2
		USN-4	Evaluate the Dataset	Data is evaluated before processing	Medium	Sprint-1
	Evaluate Accuracy of Algorithm	USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the riaht output	High	Sprint-3

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	20	High	PRIYANKA.M RAHAT SAFANA.M SNEHA.P.S SWETHA.S
		USN-2	As a user, I will receive confirmation emailonce I have registered for the application usage.	20	High	PRIYANKA.M RAHAT SAFANA.MSNEHA.P.S SWETHA.S
Sprint-2	Input	USN-3	Whenever the fire is detected, the informationis given to the database.	20	High	PRIYANKA.M RAHAT SAFANA.MSNEHA.P.S SWETHA.S
Sprint-2		USN-4	When it is the wildfire then the alarming system is activated.	20	High	PRIYANKA.M RAHAT SAFANA.M SNEHA.P.S SWETHA.S
Sprint-3	Output	USN-5	And the alarm also sent to the corresponding departments and made them know that the wildfire is erupted.	20	High	PRIYANKA.M RAHAT SAFANA.M SNEHA.P.S SWETHA.S
Sprint-4	Action	USN-6	Required actions will be taken in order to control erupted wildfire by reaching as earlyas possible to the destination with the help ofdetecting systems.	20	High	PRIYANKA.M RAHAT SAFANA.M SNEHA.P.S SWETHA.S

6.2 SPRINT DELIVERY

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 2022	05 Nov 2022	20	05 Nov 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

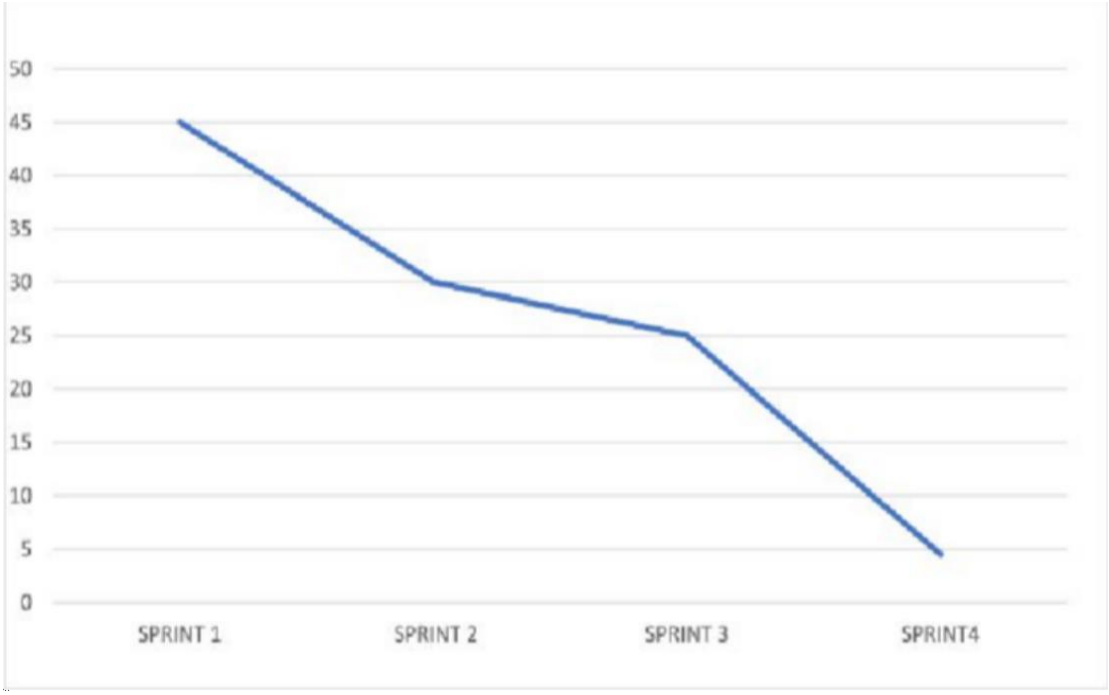
Velocity:

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) periteration unit (story points per day)

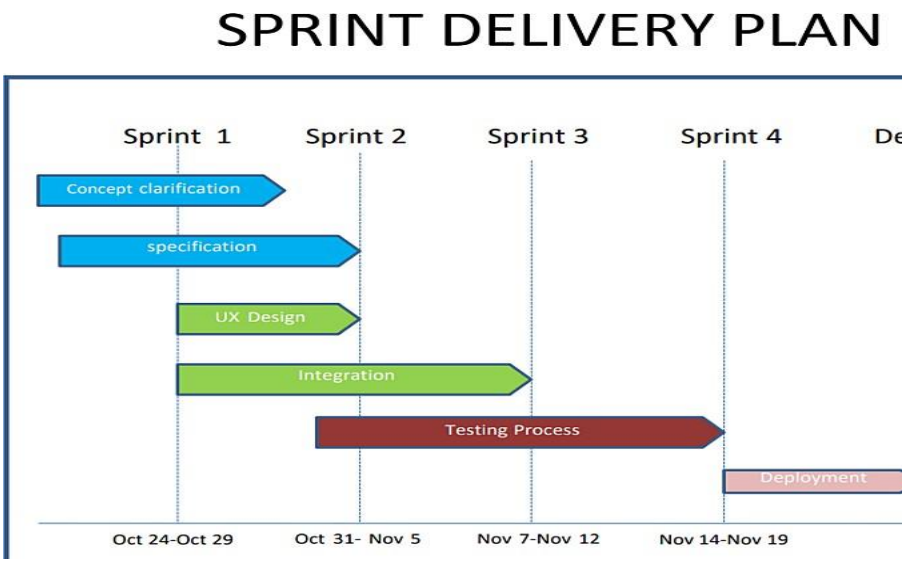
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burn down chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 REPORT FROM JIRA



JIRA has categorized reports in four levels, which are −

i. Agile
ii. Issue Analysis
iii. Forecast & Management
iv. Others

VELOCITY: SPRINT - 1

Sprint duration = 5 days
Velocity of team = 20 points

Average Velocity (AV) = $\frac{\text{Velocity}}{\text{Sprint duration}}$

AV = 20/5 = 4

Average Velocity = 4

VELOCITY: Sprint 1 - 4

Sprint duration = 20 days
Velocity of team = 80 points

Average Velocity (AV) = $\frac{\text{Velocity}}{\text{Sprint duration}}$

AV = 80/20 = 4

Total Average Velocity = 4

7. CODING AND SOLUTION

a. Feature 1

```
!pip install tensorflow
!pip install opencv-python
!pip install opencv-
contrib-pythonimport
tensorflow as tf
import numpy as np
from tensorflow
import keras
import os
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerato
from tensorflow.keras.preprocessing
import image
train=ImageDataGenerator(rescale=1./255,
                         shear_ra
                         nge=0.2,
                         rotation_
                         range=1
                         80,
                         zoom_ra
                         nge=0.2,
                         horizonta
                         l_flip=Tr
                         ue)
    train =
    ImageDataGenerator(rescale=1
    /255)test =
    ImageDataGenerator(rescale=1
    /255)
    train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/train_set",
                                 target_siz
                                 e=(128,12
                                 8),
                                 batch_size
                                 = 32,
                                 class_mod
                                 e =
                                 'binary' )
    test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/test_set",
                                 target_siz
                                 e=(128,12
                                 8),
```

```
                                batch_size
                                = 32,
                                class_mod
                                e =
                                'binary' )
        test_dataset.class_indices
        #to define linear initialisation import
        sequentialfrom keras.models import
        Sequential
        #to add layer
        import Dense from
        keras.layers import
        Dense
        #to create convolution kernel import
        convolution2Dfrom keras.layers import
        Convolution2D
        #import Maxpooling layer
        from keras.layers import
        MaxPooling2D#import
        flatten layer
        from keras.layers
        import Flattenimport
        warnings
        warnings.filterwarnin
        gs('ignore')model
        =Sequential()
#add convolutional layer from
tensorflow.keras.preprocessing import image
train=ImageDataGenerator(rescale=1./255,
                                shear_ra
                                nge=0.2,
                                rotation_
                                range=1
                                80,
                                zoom_ra
                                nge=0.2,
                                horizonta
                                l_flip=Tr
                                ue)
        train =
        ImageDataGenerator(rescale=1
        /255)test =
        ImageDataGenerator(rescale=1
        /255)
        train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/train_set",
                                target_siz
```

```
                                e=(128,12
                                8),
                                batch_size
                                = 32,
                                class_mod
                                e =
                                'binary' )
test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/test_set",
                                target_siz
                                e=(128,12
                                8),
                                batch_size
                                = 32,
                                class_mod
                                e =
                                'binary' )
test_dataset.class_indices
#to define linear initialisation import
sequentialfrom keras.models import
Sequential
#to add layer
import Dense from
keras.layers import
Dense
#to create convolution kernel import
convolution2Dfrom keras.layers import
Convolution2D
#import Maxpooling layer
from keras.layers import
MaxPooling2D#import
flatten layer
from keras.layers
import Flattenimport
warnings
warnings.filterwarnin
gs('ignore')model
=Sequential()
#add convolutional layer
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activati
on='relu')) #add maxpooling layer
model.add(MaxPooling2D(pool
_size=(2,2)))#add flatten layer
model.add(Flatten())
model.add(Dense(150,activation='relu'))

model.add(Dense(1,activation=
```

```
'sigmoid')) model.compile(loss
= 'binary_crossentropy',
            optimizer = "adam",
            metrics = ["accuracy"])
model.fit_generator(x_train,steps_per_epoch=14,epochs=5,validation_data=x_test,validation_steps=4)
model.save("/content/drive/MyDrive/archive(1)/forest1.h5")
predictions =
model.predict(test_dataset)
predictions =
np.round(predictions)
predictions
print(len(predictions))
#import load_model from
keras.modelfrom
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
#load the saved model
model =
load_model("/content/drive/MyDrive/archive(1)/forest1.h
5")def predictImage(filename):
  img1 =
  image.load_img(filename,target_size=(128,
```

```
128))Y = image.img_to_array(img1)
X =
np.expand_dims(Y
,axis=0)val =
model.predict(X)
print(val)
if val == 1:
  p
r
i
n
t
(
"
f
i
r
e
"
)
e
li
f
v
a
l
=
=
0
:
    print("no fire")
predictImage("/content/drive/MyDrive/Dataset/test_set/with fire/19464620_401.jpg")
```

b.Feature 2

```
!pip install tensorflow
!pip install opencv-python
!pip install opencv-
contrib-pythonimport
tensorflow as tf
import numpy as np
from tensorflow
import keras
import o
```

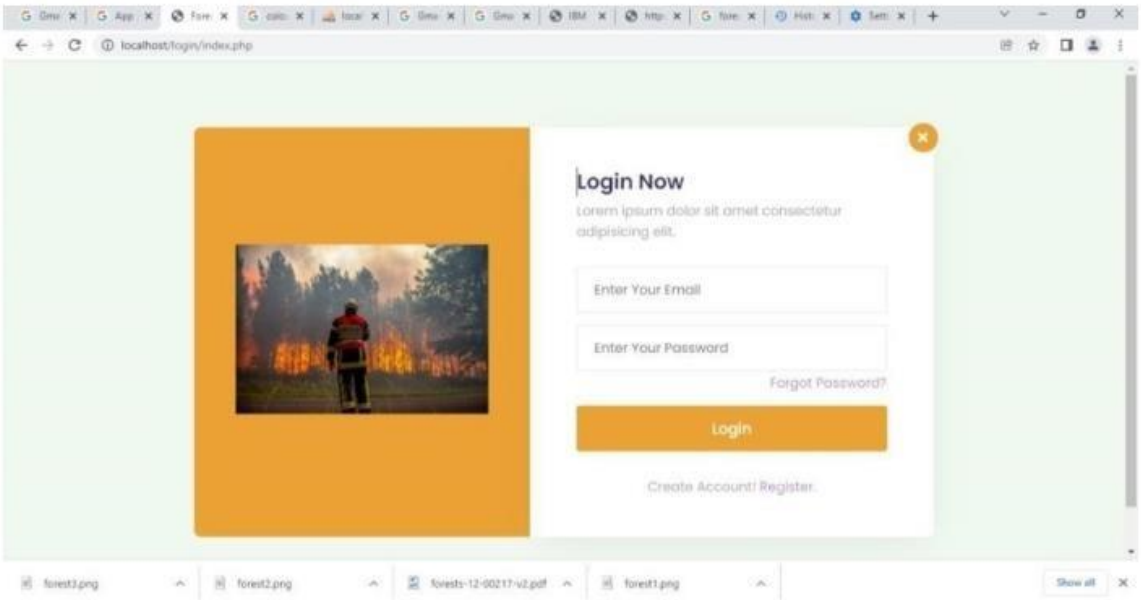
```
import cv2
        from tensorflow.keras.preprocessing.image import ImageDataGenerator
        from tensorflow.keras.preprocessing import image
        train=ImageDataGenerator(rescale=1./255,
                                 shear_range=0.2,
                                 rotation_range=180,
                                 zo om_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/Dataset/train_set",
                                          target_size=(128,128),
                                          batch_size = 32,
                                          class_mode = 'binary' )
        test_dataset = test.flow_from_directory("/content/drive/MyDrive/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 =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())
        model.add(Dense(150,activation='relu'))


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

```
        model.compile(loss = 'binary_crossentropy',
                      optimizer = "adam",
                      metrics = ["accuracy"])
        model.fit_generator(x_train,steps_per_epoch=14,epochs=5,validation_data=x_test,validation_steps=4)
        model.save("/content/drive/MyDrive/archive(1)/forest1.h5")
        predictions = model.predict(test_dataset)
        predictions = np.round(predictions)
        predictions
        print(len(predictions))
        #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
        #load the saved model
        model = load_model("/content/drive/MyDrive/archive(1)/forest1.h5")
        def predictImage(filename):
          img1 = image.load_img(filename,target_size=(128,128))
          Y = image.img_to_array(img1)
          X = np.expand_dims(Y,axis=0)
          val = model.predict(X)
          print(val)
          if val == 1:
            print(" fire")
          elif val == 0:
              print("no fire")
        predictImage("/content/drive/MyDrive/Dataset/test_set/with fire/19464620_401.jpg")
pip install twilio
pip install playsound
#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
```

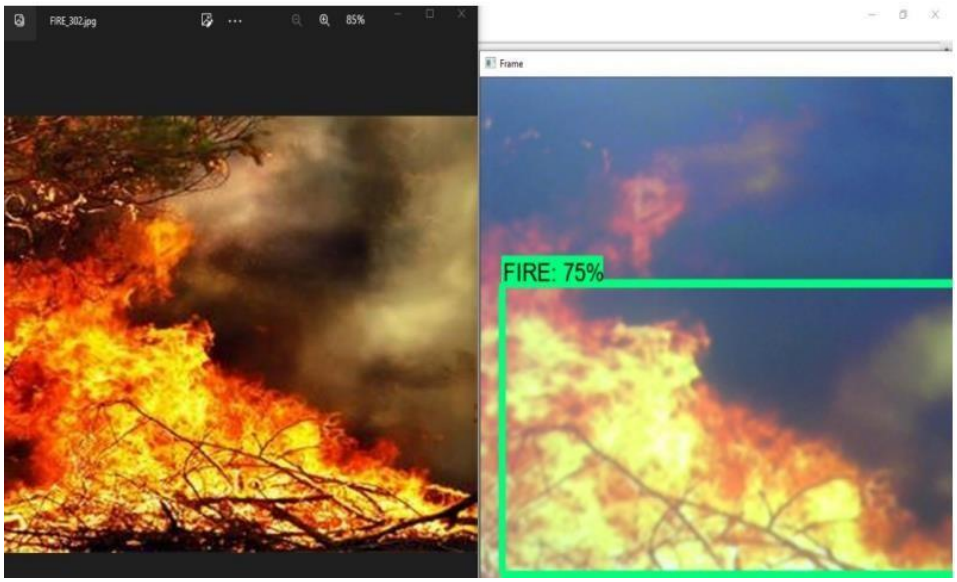
```
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
#load the saved model
model = load_model(r'/content/drive/MyDrive/archive(1)/forest1.h5')
#define video
video = cv2.VideoCapture('/content/Fighting Fire with Fire _ Explained in 30 Seconds.mp4')
#define the features
name = ['forest','with forest']
account_sid='ACfb4e6d0e7b0d25def63044919f1b96e3'
auth_token='f9ae4fc4a617a527da8672e97eefb2d8'
client=Client(account_sid,auth_token)
message=client.messages \
.create(
     body='Forest Fire is detected, stay alert',
     from_='+1 302 248 4366',
     to='+91 99400 12164'
)
print(message.sid)
pip install pygobject
def message(val):
  if val==1:
    from twilio.rest import Client
    print('Forest fire')
    account_sid='ACfb4e6d0e7b0d25def63044919f1b96e3'
    auth_token='f9ae4fc4a617a527da8672e97eefb2d8'
    client=Client(account_sid,auth_token)
    message=client.messages \
     .create(
        body='forest fire is detected, stay alert',
        #use twilio free number
        from_='+1 302 248 4366',
        #to number
        to='+91 99400 12164')
    print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
  elif val==0:
    print('No Fire')
from matplotlib import pyplot as plt
```

```
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
img1 = image.load_img('/content/drive/MyDrive/Dataset/test_set/with
fire/Wild_fires.jpg',target_size=(128,128))
Y = image.img_to_array(img1)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img1)
plt.show()
message(val)
img2 =
image.load_img('/content/drive/MyDrive/Dataset/test_set/forest/1200px_Mountainarea.jpg',target_size=(128,1
28))
Y = image.img_to_array(img2)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img2)
plt.show()
message(val)
```

8.TESTING



9.RESULTS



Value obtained from three sensor, if any Infrared ray detected, it gives output as IRdetected, Sensor activated! Similarly, if there is any temperature change it will show Abnormal temperature and its intensity. For any smoke detection it output as Smoke detected and sensor value.Above image is result obtained from the trained ML model showing count for damaged and intact homes.

10. ADVANTAGES & DISADVANTAGES

It refreshes the habitat zones: Fire clears out plants and trees to make more natural resources available to the habitat. Fewer trees mean more water becomes available for the remaining plants and animals that call the area their home. New grass and shrubs are food sources for a number of animals as well. A ground cover that comes back after a fire becomes a new micro-habitat. Everything isrefreshed with a fire.

1. **Low-intensity fires don't usually harm trees:** The bark of a tree is like an armored shell against fire, pests, and other things that could damage them. Most forest fires burn at low- temperature levels when conditions are optimal and this causes minimal damage to the trees of the forest when it occurs. The end result is a clearing of the ground floor of the forest while the trees are able to continue standing majestically.

Decreases the Wastes on Forests: Forests have a lot of waste that ends up building up over time and these wastes can help create wildfires. If a large wildfire breaks out it might take weeks to control it and the damage it can cause is just too extensive to understand for us. Waste such as dead leaves on the ground can be pretty useful for wildfires to feed on and small forest fires just deal with these wastes properly without going out of control.

Disadvantages:

1. **A forest fire sets up the potential for soil erosion to occur:** Forest fires clear the underbrush away and encourage new growth, but there is a period of time between the fire and the new growth where the forest is vulnerable.

Forest fires always bring death in some form: Maybe it's just the weak plants of the forest that are killed during a fire, but there is always some sort of death that happens when a fire occurs. Sometimes it is the firefighters who are tasked with stopping the fire. It could be animals or pets.

1. **Uncontrolled fires can cause localized air pollution:** Despite the amount of global development that has occurred, there are many forests that are difficult or nearly impossible to reach. Fires in these areas are left to burn in an uncontrolled fashion and this creates air pollution which can affect the local environment and make it difficult to breathe.

1. CONCLUSION

This project will help in early detection of forest fire and the prevention. It also involves the risk factor of analyzing the drone images of affected areas using machine learning algorithm which overcomes the existing project. This system detects the fire conditions in a short time before any fire accidents spreads over the forest area. The scope of using video frames in the detection of fire using machine learning is challenging as well as innovative. If this system with less error rate can be implemented at a large scale like in big factories, houses, forests, it is possible to prevent damage and loss due to random fire accidents by making use of the Surveillance System.

2. FUTURE SCOPE

Future Scope In future, we are planning to install smart water tank system in dense forest where reachability of resources and firefighters is difficult. In addition to that we will be updating the system with more features and reliability. We will also include a highpitch sound system that will keep away the animals from the site of fire.The proposed system can be developed to more advanced systemby integrating wireless sensors with CCTV for added protection and precision. The algorithm shows great promise in adapting tovarious environment.

DEMO LINK:

https://www.youtube.com/embed/tLLSVqQYB_A

GITHUB: IBM-EPBL/IBM-Project-11300-1659287127: Emerging Methods for Early Detection of Forest Fires (github.com)