Team ID	PNT2022TMID06675
Team Leader	S.NIVETHA
Team Members	M.MANJU MITHRA
	V.PAVITHRA
	P.SHARUHASHINI

EMERGING METHODS OF EARLY DETECTION OF FOREST FIRES

AGENDA

13.APPENDIX

13.1 Source Code

13.2 GitHub & Project Demo Link

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Traditional forest fire detection methods include satellite monitoring, ground patrols, watch towers, among others, which have high labor and financial costs in return for low efficiency. Current remote sensor technologies are becoming more common, but primarily rely on battery technology for power.

1.2 PURPOSE

Emerging Methods for Early Detection of Forest Fires The project aims to build a model that detects forest fires using convolutional neural networks.

2. LITERATURE SURVEY

2.1 Existing Problem

It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency.

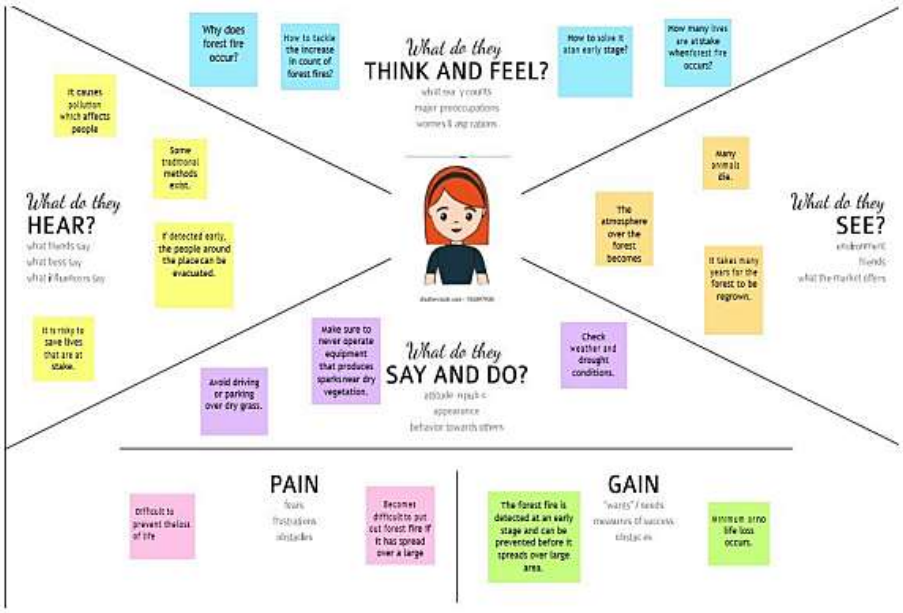
2.2 References

https://www.researchgate.net/publication/334418384_Early_Forest_Fire_Detection_Using_Drones_and_Artificial_Intelligence

3. IDEATION & PROPOSED SOLUTION

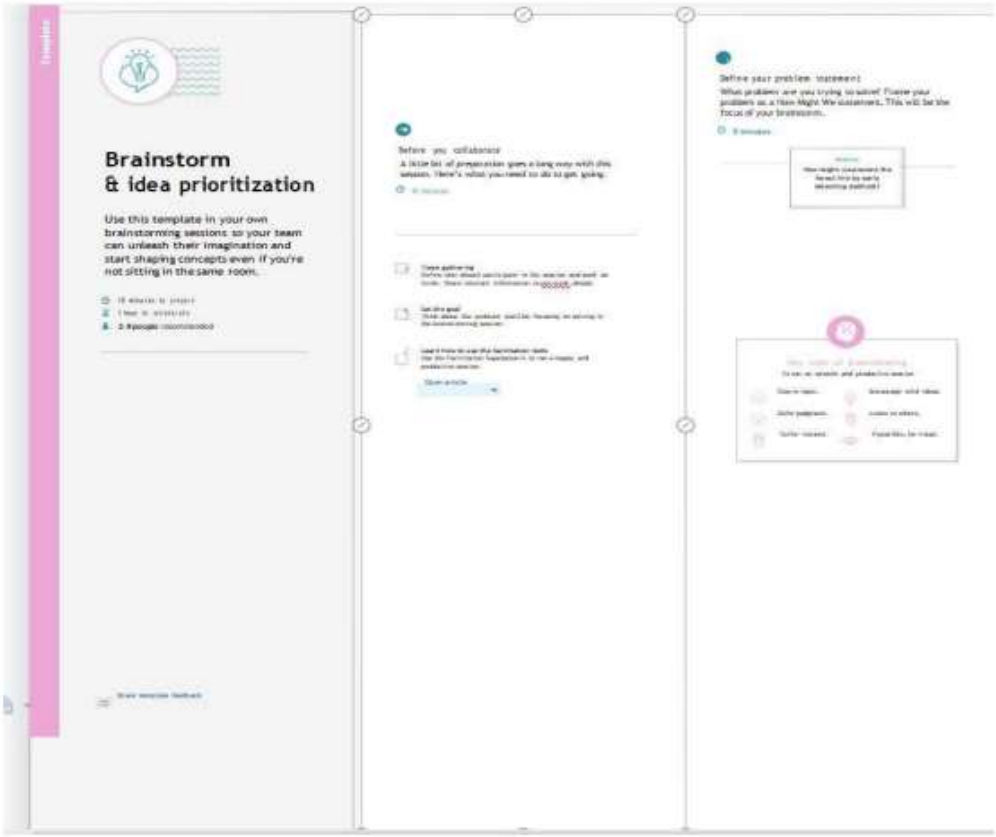
3.1 Empathy Map Canvas

Empathy Map:

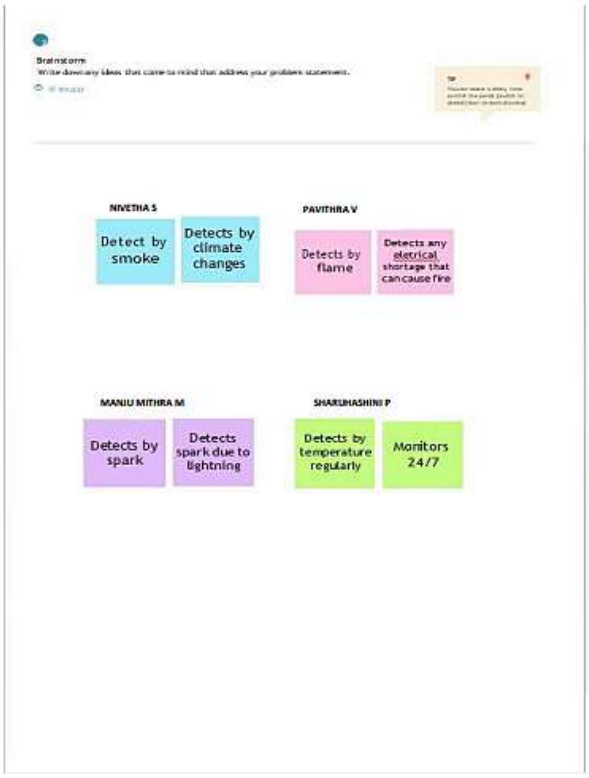


3.2Ideation&Brainstroming

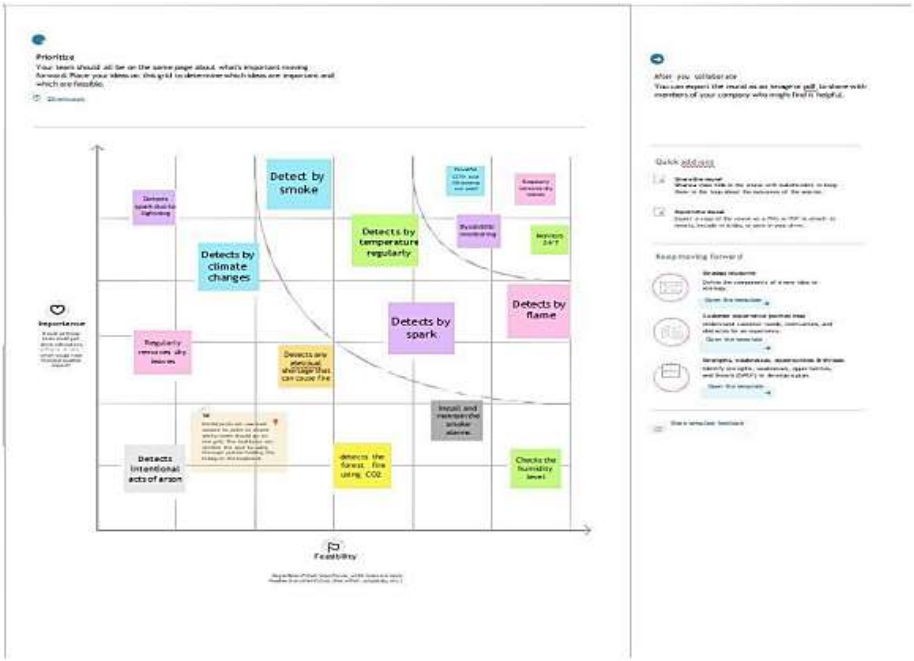
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

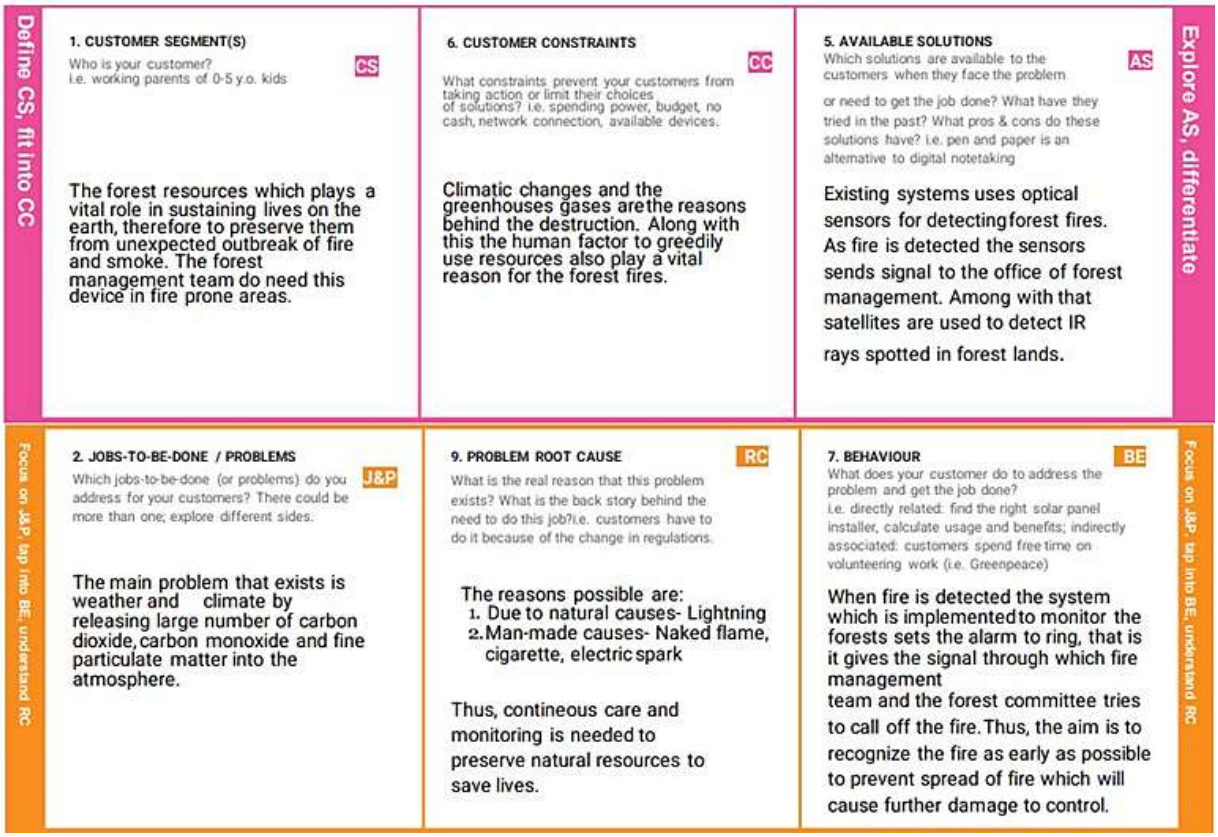


Proposed solution

s.no	Parameter	description
1.	Problem Statement (Problem to be solved)	A forestfire risk prediction algorithm, based onsupport vector machines, is presented. The algorithm depends on previous weather conditions in order to predict the fire hazard level of a day.
2.	Idea / Solution description	Use computer vision methods for recognitionand detection of smoke or fire.
3.	Novelty / Uniqueness	Real time computer programdetect forest firein earliestbefore it spreadto larger area
4.	Impact on society	Blocked roads and railwaylines, electricity, mobile and land telephone lines cut,destruction of homes and industries.
5.	Business Model (Revenue Model)	The proposed method was implemented using the Python programming language on a Core i3or greater(CPU and 4GB RAM.)
6.	Scalability of the Solution	Computer visionmodeenable land coverclassification and smoke detectionfrom satellite and groundcameras

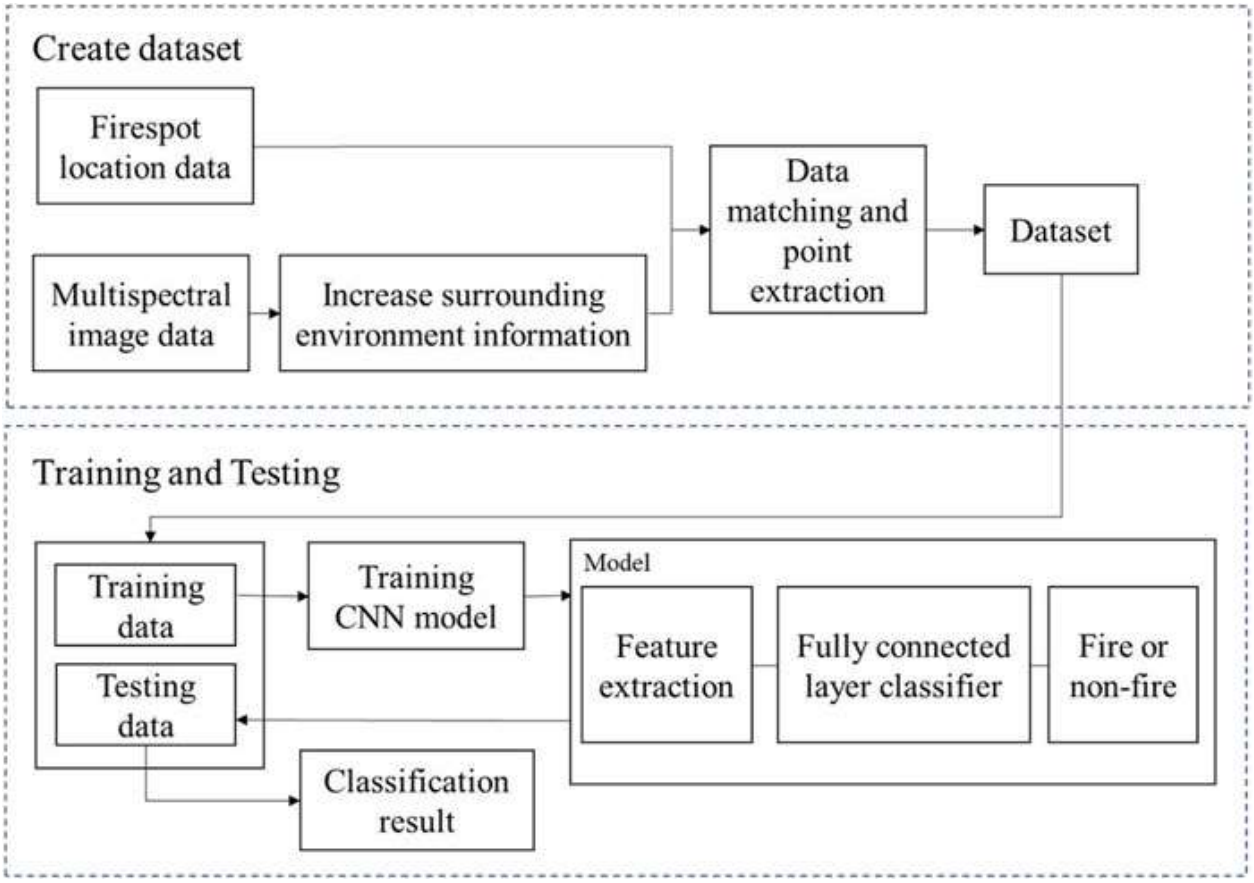
Problem Solution fit

Project Title: Emerging Methods for early detection of forest fire **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID15454



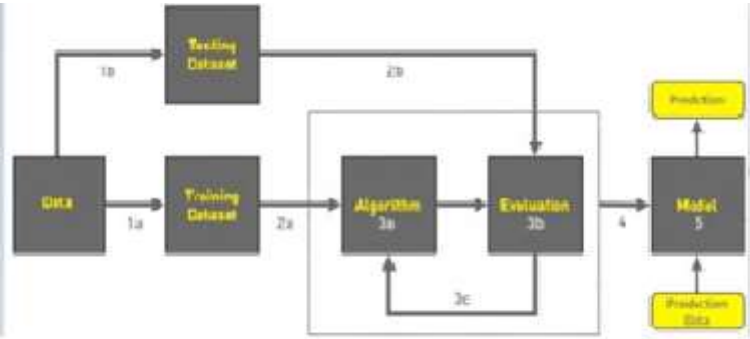
4. REQUIREMENT ANALYSIS

4.1 Functional requirement

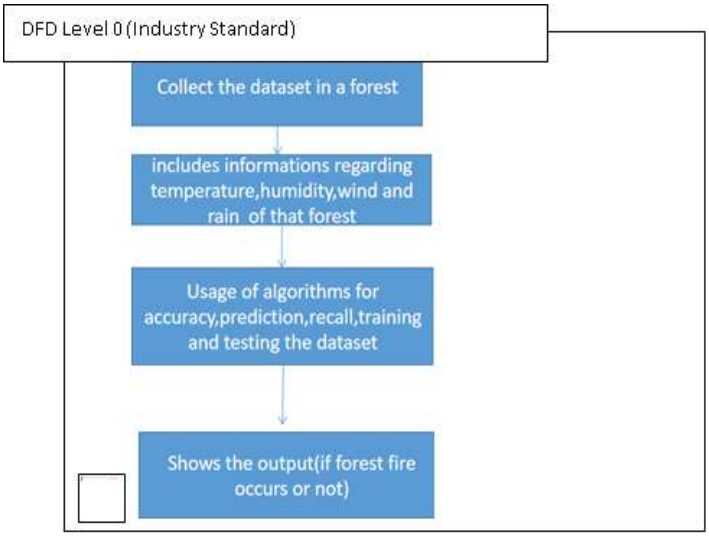


5. PROJECT DESIGN

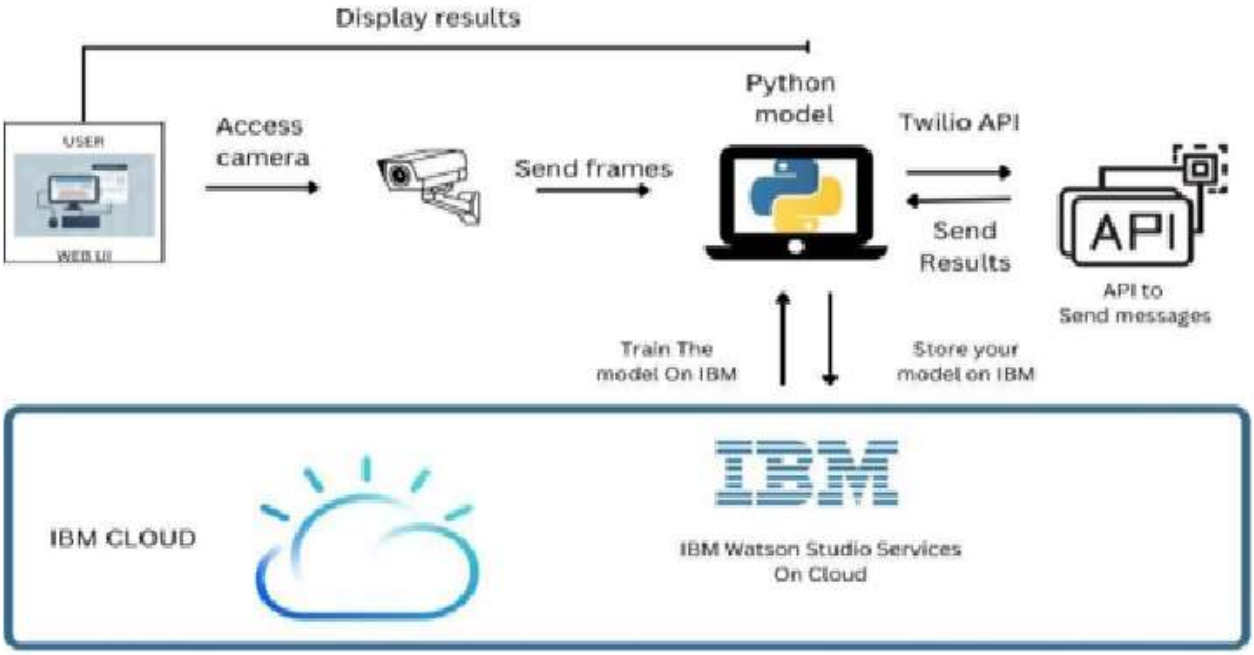
5.1 Data Flow Diagrams



1. COLLECT DATA
2. EVALUATE DATASET
3. IMPLEMENT ALGORITHMS
4. EVALUATE THEACCURACYOF EACH ALGORITHMS



5.2 Solution & Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance criteria	Priority	Release
Environmentalist	Collect the data	USN-1	As an Environmentalist,it is necessary to collect thedata of theforest whichincludes temperature, humidity,wind and rain of theforest	It is necessary to collectthe right data else the prediction may becomewrong	High	Sprint-1
		USN-2	Identify algorithms that can be used forprediction	To collect the algorithm to identify theaccuracy level of each algorithms	Medium	Sprint-2
		USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtainthemost accurate output	High	Sprint-2
		USN-4	Evaluate the Dataset	Data is evaluated beforeprocessing	Medium	Sprint-1
		USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining theright output	High	Sprint-3
		USN-6	Outputs fromeachalgorithm are obtained	It is highly used to predictthe effect and to take precautionary measures.	High	Sprint-4

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	Collect Dataset	20	*High*	*Nivetha.S* *Manju mithra.M* *Pavithra.V* *Sharuhashini .P*
Sprint-1		USN-2	Image preprocessing	20	*High*	*Nivetha.S* *Manju mithra.M* *Pavithra.V* *Sharuhashini .P*
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessarylayersand compile the mode	20	*High*	*Nivetha.S* *Manju mithra.M* *Pavithra.V* *Sharuhashini .P*
Sprint-2		USN-4	Training theimage classification model using CNN	20	High	Nivetha.S Manju mithra.M Pavithra.V Sharuhashini .P

Sprint-3	Training and Testing	USN-5	Training the model and testing the model'sperformance	20	High	Nivetha.S Manju mithra.M Pavithra.V Sharuhashini .P
Sprint-4	Implemen tation of theapplica tion	USN-6	When it is the wildfire then the alarmingsystem is activated. And the alarm will be sent to the corresponding department and required action will be taken soon to control the fire.	20	High	Nivetha.S Manju mithra.M Pavithra.V Sharuhashini .P

6.2 Sprint Delivery Schedule:

Sprint	**Total story points**	**Duration**	**Sprint StartDate**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint ReleaseDa te(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

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

7.1 Feature 1:

We uploaded the dataset that is given and have divided the classes into train set and data set and preprocessed the image.The output is shown here.

In [3]:
```
#Applying ImageDataGenerator functionality to trainset
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/train_set',target_size=(128,128),batch_size=32,class_mode='binary')
```

```
Found 439 images belonging to 2 classes.
```

In [4]:
```
#Applying ImageDataGenerator functionality to testset
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.
```

7.2 Feature 2:

After the image preprocessing we have done the model building.The model building output is shown here.

In [27]:
```
#Load the saved model
model = load_model("forest1.h5")
```

In [29]:
```
img=image.load_img('/content/drive/MyDrive/Dataset/test_set/with fire/180802_CarrFire_010_large_700x467.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
```

In [31]:
```
pred=model.predict(x)
```

```
1/1 [==============================] - 0s 31ms/step
```

In [32]:
```
pred
```

Out[32]:
```
array([[1.]], dtype=float32)
```

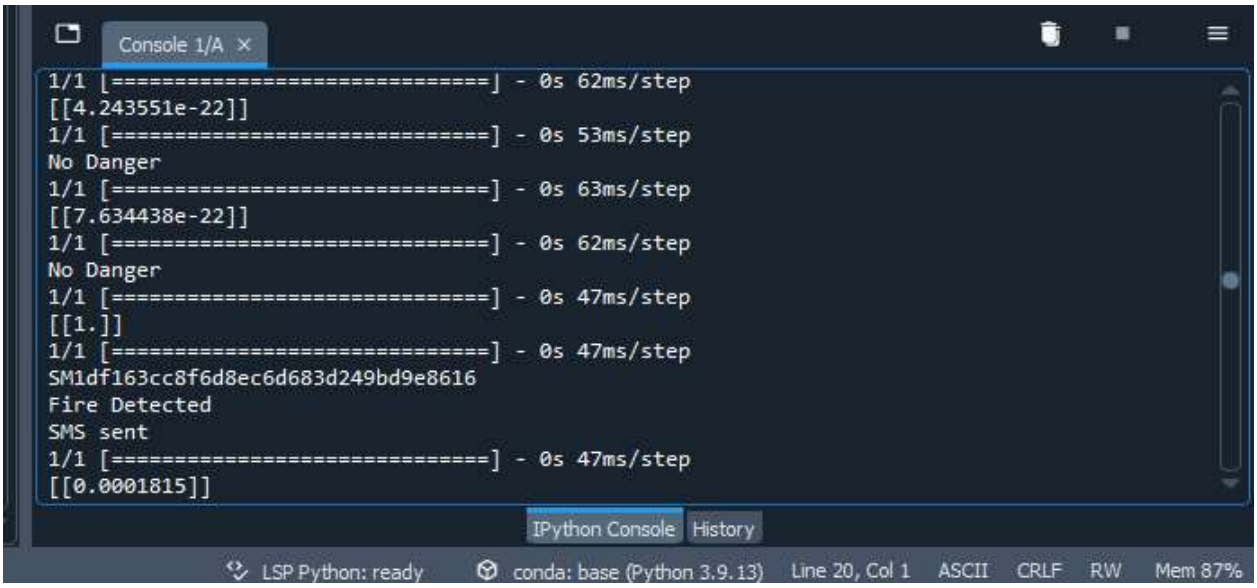
In [ ]:
```

```

By using the above forest1.h5 model we can take our desired output according to the input.

8. TESTING

8.1 Test Cases:

By the showing image of forest fire the desired output of "Forest fire is detected,stay alert" is sent via SMS form twilio service.By showing the image of forest the desired output is no danger.



8.2 User Acceptance Testing:

We have tested our project by showing the image of forest with fire and forest without fire.The output is shown above.

9. RESULTS

9.1 Performance Metrics:

Model evaluation

In [27]:
```
#Load the saved model
model = load_model("forest1.h5")
```

In [29]:
```
img=image.load_img('/content/drive/MyDrive/Dataset/test_set/with fire/180802_CarrFire_010_large_700x467.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
```

In [31]:
```
pred=model.predict(x)
```

```
1/1 [==============================] - 0s 31ms/step
```

In [32]:
```
pred
```

Out[32]:
```
array([[1.]], dtype=float32)
```

In [ ]:

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Avoid Smoke Inhalation. The most important reason is perhaps the only one you really need.
2. Early Detection. The earlier a fire is detected, the faster it will be that firefighters will respond.
3. Insurance Discounts.
4. 24/7 Monitoring.
5. Easy & Affordable.

DISADVANTAGES:

1. The system is essentially useless if the batteries aren't charged, since it won't work properly.
2. There is a bit of a burden to business owners to always remember to keep the batteries fresh so the system operates properly when you need it most.

11. CONCLUSION

Early fire detection is best achieved by the installation and maintenance of fire detection equipment in all areas of the forest.

12. FUTURE SCOPE

The future will be with multicriteria detection in which the detector will be more of a sensor, with the detection more for the products of combustion, such as carbon monoxide, carbon dioxide, sulfur dioxide, nitrogen oxides in addition to heat and particulate matter.

13. APPENDIX

Source Code: Python code

```
#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'forest1.h5')
#define video
video = cv2.VideoCapture(0)
#define the features
name = ['forest','with forest']

account_sid = 'AC557b4c7a685d072baa73125f61031af3'
auth_token = 'a59cd5e5fdfddcc9ab008273557f8f78'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
        body='Forest fire is detected , stay alert',
        from_='+14247991869',
        to='+918940722793'
    )

print(message.sid)
```

```
#import opencv library
import cv2
#import numpy
import numpy as np
#import images and load_model function from keras
from keras_preprocessing import image
from keras.models import load_model
#import client from twilio API
from  twilio.rest import Client
#import playsound package
from playsound import playsound

#load the saved model
model = load_model(r'forest1.h5')
video = cv2.VideoCapture(0)
name = ['forest','with fire']

while(1):


    success, frame=video.read()
    cv2.imwrite("image.jpg",frame)
    img=image.load_img("image.jpg",target_size=(128,128,3))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    pred=model.predict(x)
    p=pred[0]
    print(pred)
    ##cv2.putText(frame,"predicted class= "+str(name[p]), (100,100),
      ##          cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
    pred=model.predict(x)


    if pred[0]==1:


        account_sid = 'AC557b4c7a685d072baa73125f61031af3'
```

```
        auth_token = 'a59cd5e5fdfddcc9ab008273557f8f78'
        client = Client(account_sid, auth_token)
        message=client.messages\
        .create(
        body='Forest Fire is Detected, stay alert',
        from_='+14247991869',to='+918940722793')

        print(message.sid)
        print('Fire Detected')
        print('SMS sent')
        playsound(r'C:\Users\My\Downloads\buzzer.mp3')

    else:
        print("No Danger")

        cv2.imshow("image",frame)

    if cv2.waitKey(1) & 0xFF ==ord('a'):
        break
video.release()
cv2.destroyAllWindows()
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-30013-1660138238

Project Demo Link:

https://www.youtube.com/embed/97uqCIgVIvI