Project Name:
Emerging Methods for Early Detection of Forest Fires

Team ID : PNT2022TMID32964

Team Size : 4

Team Leader : VINOTH KUMAR M

Team member : SUDHARSAN S

Team member : VENKADA SUBRAMANIAN D

Team member : VEERENTHIRAN S

Project Report

Project Overview:

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. 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. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities.

Purpose:

Due to delayed detection of fire in the Forest,there is a heavy loss of both flora and fauna.It also pollutes the already polluted atmosphere.Many rare species of animals and birds are lost in the river of time.Precious herbs are damaged.

To prevent all the above from happening we have developed this project using Convolutional Neural Network.

LITERATURE SURVEY

Existing problem

Forest fires are disasters that cause extensive damage to the entire world in economic, ecological, and environmental ways. These fires can be caused by natural reasons, such as high temperatures that can create spontaneous combustion of dry fuel such as sawdust, leaves, lightning, etc., or by human activities, such as unextinguished campfires, arson, inappropriately burned debris, etc1. According to research, 90% of the world's forest fire incidents have occurred as a result of the abovementioned human carelessness1. The increase in carbon dioxide levels in the atmosphere due to forest fires contributes to the greenhouse effect and climate change. Additionally, ash destroys much of the nutrients in the soil and can cause erosion, which may result in floods and landslides.

At earlier times, forest fires were detected using watchtowers, which were not efficient because they were based on human observations. In recent history and even the present day, several forest fire detection methods have been implemented, such as watchtowers, satellite image processing methods, optical sensors, and digital camera-based methods2, although there are many drawbacks, such as inefficiency, power consumption, latency, accuracy and implementation costs. To address these drawbacks, a forest fire detection system using wireless sensor networks is proposed in this paper.

References

- Surapong Surit, Watchara Chatwiriya. Forest Fire Smoke Detection in Video Based on Digital Image Processing Approach.
- Osman Gunay, A. Enis C ̧ Etin, Yusuf Hakan, Habiboglu. Flame Detection method in video using Covariance descriptors, IEEE transactions, 1817-1820.
- Yusuf Hakan Habibo ̆glu, Osman Günay, A. Enis Çetin (2011). Covariance matrix-based fire and flame detection method in video", Springer-Verlag, 17.
- CHENG Caixia, SUN Fuchun, ZHOU Xinquan (2011). One Fire Detection Method Using Neural Networks, Tsinghua Science and Technology, ISSN 1007-0214 05/17 31-35Volume 16, Number 1.

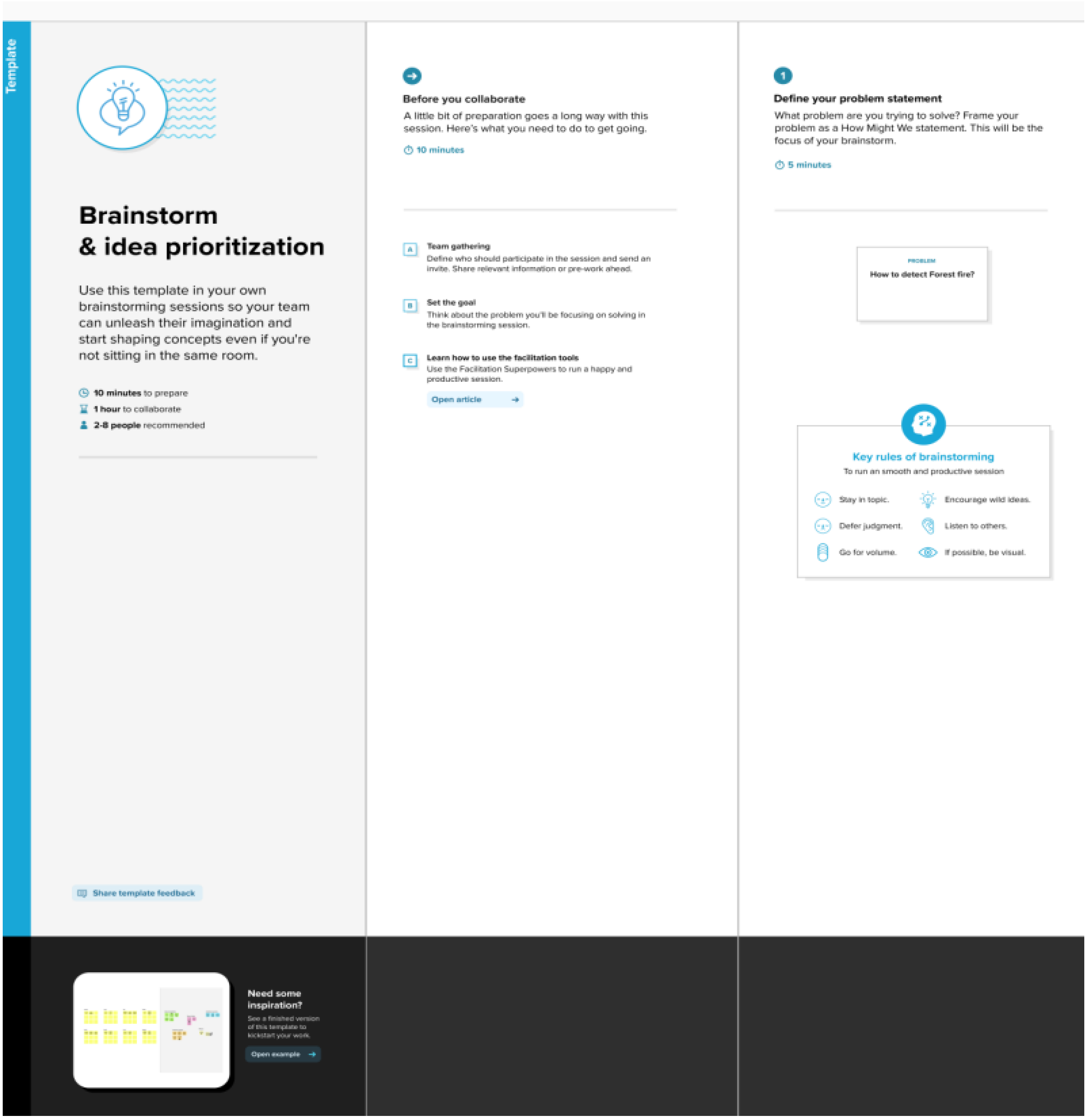
Problem Statement Definition

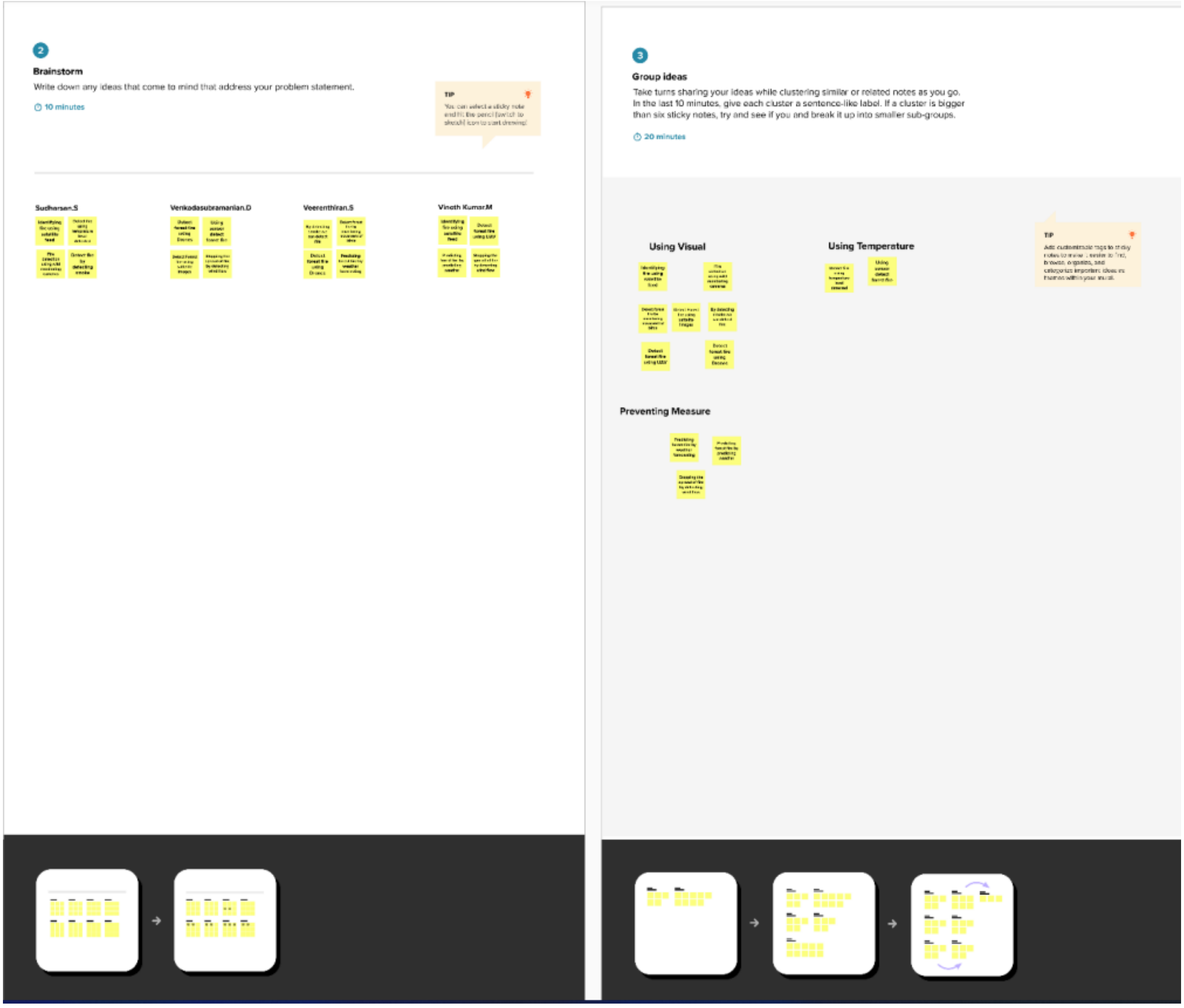
Fire detection at an early stage is important for the safety of the people. Lack of information due to manual detection is the main cause of failure of fire detection. Fire can be detected by using smoke at an early stage as it is the fire indicator. Generally automatic forest fire detection using image processing techniques represents one of the significant aspects of forest fire avoidance earlier. Detection using image and video is effective than using sensors. In image processing the inputs for the fire detection may be an image or a video but the input as a video is quite complex process but provides good result. The techniques such as Wavelet decomposition, spatial and temporal analysis, Gaussian Mixture Model, Multi-Feature fusion detect fire in an accurate manner.

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

Ideation & Brainstorming



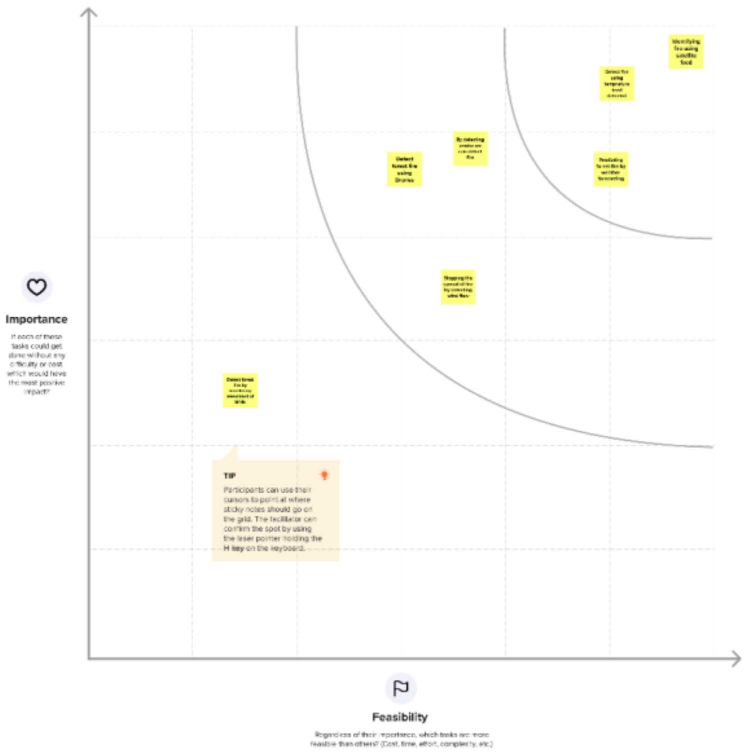

2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!
Sudharsan.S
Venkadasubramanian.D
Veerenthiran.S
Vinoth Kumar.M
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go.
In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger
than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Using Visual
Using Temperature
TIP
Preventing Measure

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



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

Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Detecting forest fires in an early stage to avoid massive damage.
2.	Idea / Solution description	Identifying huge forest fires in real-time utilising AI algorithms with camera and satellite footage. The systems then notify dispatchers and local authorities about the new ignition
3.	Novelty / Uniqueness	Convolutional Neural Network system allows us to deliver information more quickly and accurately. It is possible to deploy a comprehensive coverage, which is nearly impossible.
4.	Social Impact / Customer Satisfaction	Monitoring of the potential danger regions and early identification of fire can greatly minimise the response time, as well as potential damage and firefighting expenses, while also saving many lives.
5.	Business Model (Revenue Model)	Subscription Model
6.	Scalability of the Solution	Despite the physical distance between resources and users, its regionally scalable system maintains its usability and utility;

Problem Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Forest Officers who need to avoid forest fires to prevent damage of wildlife.

6. CUSTOMER CONSTRAINTS CC

Due to low Budget
Convenience of chosen methods
Lack of Man power

5. AVAILABLE SOLUTIONS AS

Manual detection of Forest fire using human site
Pros:Lack of complexity
Cons:Continuous monitoring is not possible,
Delay in detection

Explore AS, differentiate

Focus on J&P, tap into BE,

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

Identify the place where forest fired

Forest fire will lead to many wildlife damages

9. PROBLEM ROOT CAUSE RC

Lack of continuous monitoring and techniques to detect fire immediately

7. BEHAVIOUR BE

directly related: Assigning human resources to monitor the forest
indirectly associated: Monitoring weather reports periodically

Identify strong TR & EM

Forest Officers finds more helpful than manual analyzing

Forest Officers predicted the Forest fire before it happens Which trigger them to use this software

Detection and notification of forest fire through video feed obtained by UAV,Wildlife monitoring cameras or satellite
Fire is detected by using image processing done by neural networks

8.1 ONLINE
Weather monitoring apps through Internet

8.2 OFFLINE
Human interaction through mobiles,face to face

Identify strong TR & EM

4. EMOTIONS: BEFORE / AFTER EM

Inaccuracy>Accuracy
Forest fire not controlled>forest fire controlled

REQUIREMENT ANALYSIS

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	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login using credentials
FR-4	User Profile	User details to be displayed
FR-5	User Application	User is alerted if there is a forest fire occurrence in their surroundings

Non-functional Requirements:

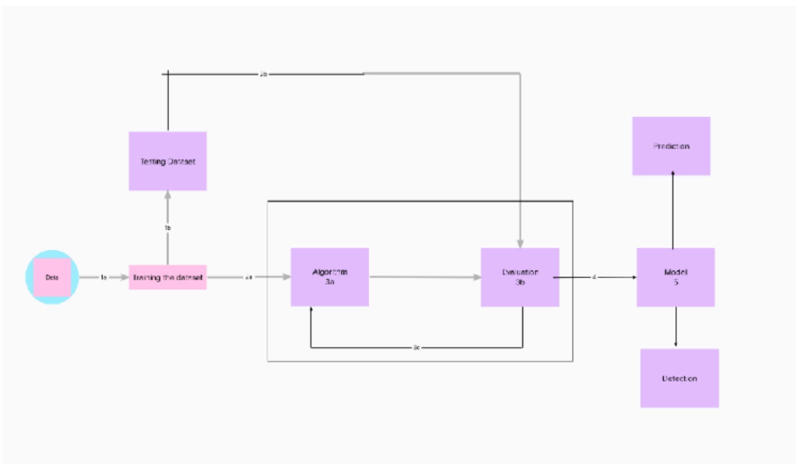
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Alerts user if there is fire in given location
NFR-2	Security	Instant live feed with alert of the situation
NFR-3	Reliability	The prediction of the forest fire is 87% accurate
NFR-4	Performance	The feed and the alert message action is without a lag
NFR-5	Availability	The application gives alerts and live feeds 24/7
NFR-6	Scalability	Early detection and alerting users are done efficiently and in a faster means

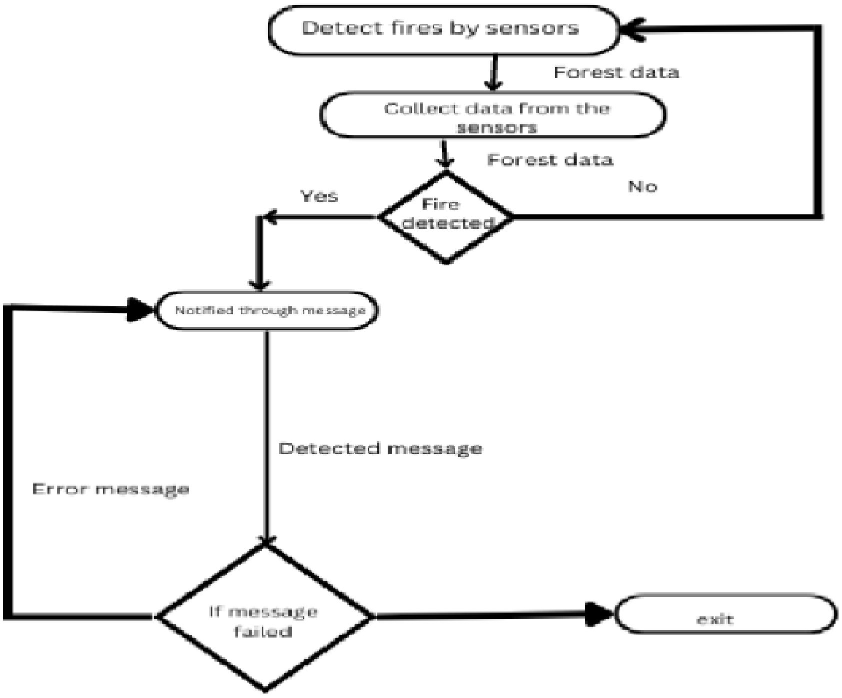
PROJECT DESIGN

Data Flow Diagrams:

DATA FLOW DIAGRAMS:

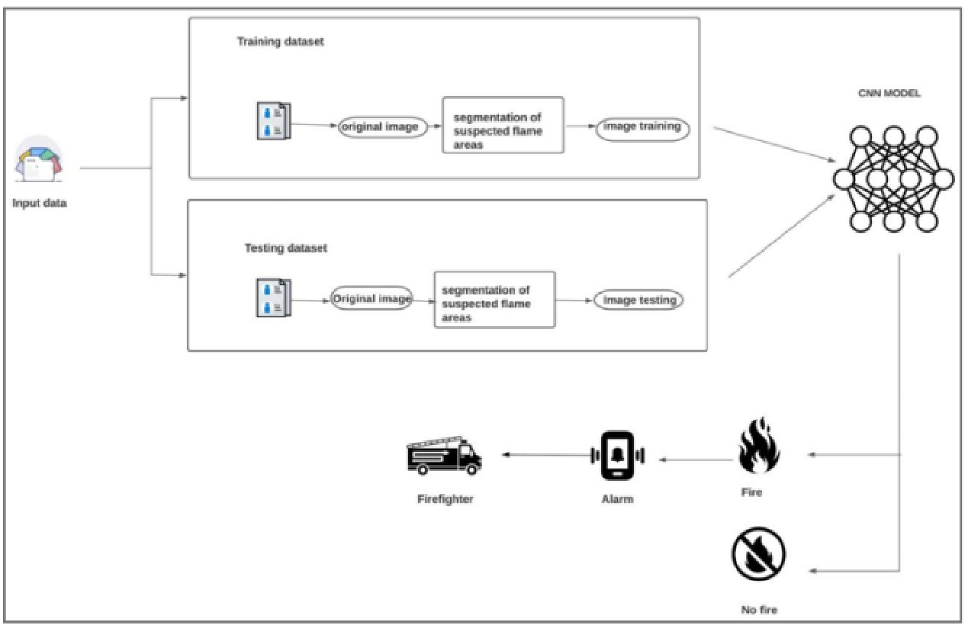


1. COLLECT DATA
2. EVALUATE DATA SET
3. IMPLEMENT ALGORITHMS
4. EVALUATE THE ACCURACY OF EACH ALGORITHMS
5. DISPLAY RESULTS



Solution & Technical Architecture

Solution Architecture Diagram:



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 the data of the forest which includes temperature ,humidity, wind and rain of the	It is necessary to collect the right data else the prediction may become	High	Sprint-1
		USN-2	Identify algorithm that can be used for prediction	To collect the algorithm to identify the accuracy 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 obtain the most accurate output	High	Sprint-2
		USN-4	Evaluate the dataset	Data is evaluated before processing	Medium	Sprint-1
		USN-5	Identify accuracy, precision, recall of each algorithms	These values are important for obtaining the right output	High	Sprint-3
		USN-6	Outputs from each algorithm are obtained	it is highly used to predict the effect and to take precautionary measures.	High	Sprint-4

Project Planning And Scheduling

Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset Collection	USN-1	To collect the data from various sources such as surveillance cameras or drone cameras which are used for observation of forest areas.	10	High	Vinoth Kumar .M , Venkadasubramanian.P
Sprint-1	Image pre-processing & Training the model.	USN-2	Sort and classify the collected data and process those data by training and testing the data using CNN model.	10	High	Sudharsan.S, Venkadasubramanian.P
Sprint-2	Evaluation of the model	USN-3	Evaluate the model to check whether it works efficiently and with high performance with low failure rates.	20	Medium	Veerenthiran.S, Vinoth Kumar .M
Sprint-3	Testing the model	USN-4	To test the model with the intent to find whether its satisfies the specified requirements or not	20	High	Veerenthiran.S, Sudharsan.S,
Sprint-4	Deployment	USN-5	After testing, the model is implemented on the user's offices for further use.	20	High	Vinoth Kumar .M , Venkadasubramanian.P

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 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

Reports from JIRA

Reporting helps you track and analyse your team's work throughout a project. Jira Software has a range of reports that you can use to show information about your project, versions, epics, sprints, and issues.

14. CODING & SOLUTIONING

Feature 1:

Real time Detection of Forest Fire

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

```
Mounted at /content/drive
```

In [ ]:

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

In [ ]:

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

```
test = ImageDataGenerator(rescale=1/255)
```

In [ ]:

```
train_dataset                                                                  =
train.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.

In [ ]:

```
test_dataset                                                                   =
test.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.

In [ ]:

```
test_dataset.class_indices
```

Out[ ]:

{'forest': 0, 'with fire': 1}

In [ ]:

```
#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')
```

In [ ]:

```
model = keras.Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='rel
u'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

In [ ]:

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

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

In [ ]:

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

In [ ]:

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

```
Epoch 1/5
14/14 [==============================] - 184s 13s/step - loss: 0.6098 -
accuracy: 0.6261 - val_loss: 0.5014 - val_accuracy: 0.7273
Epoch 2/5
14/14 [==============================] - 25s 2s/step - loss: 0.3824 -
accuracy: 0.8417 - val_loss: 0.0858 - val_accuracy: 0.9752
Epoch 3/5
14/14 [==============================] - 27s 2s/step - loss: 0.3452 -
accuracy: 0.8349 - val_loss: 0.2229 - val_accuracy: 0.9008
Epoch 4/5
14/14 [==============================] - 26s 2s/step - loss: 0.2595 -
accuracy: 0.8853 - val_loss: 0.1087 - val_accuracy: 0.9917
Epoch 5/5
14/14 [==============================] - 27s 2s/step - loss: 0.2018 -
accuracy: 0.9243 - val_loss: 0.0621 - val_accuracy: 0.9917
```

In [ ]:

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

```
4/4 [==============================] - 5s 1s/step
```

In [ ]:

```
predictions
```

In [ ]:

```
print(len(predictions))
```

```
121
```

In [ ]:

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

In [ ]:

```
#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
```

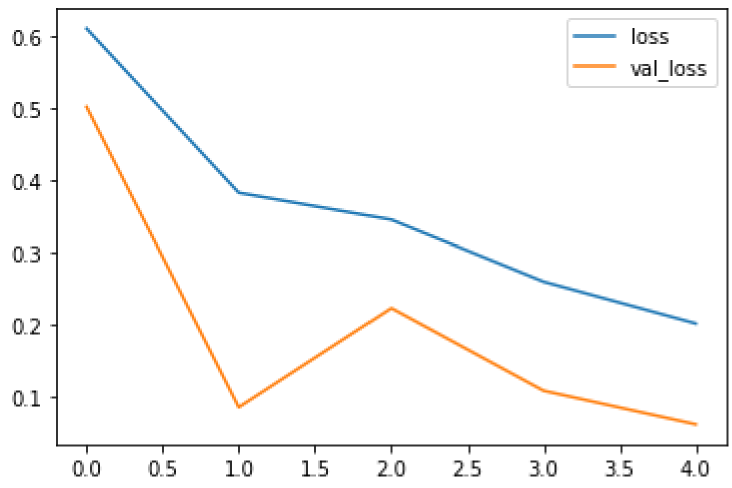
In [ ]:

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

In [ ]:

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

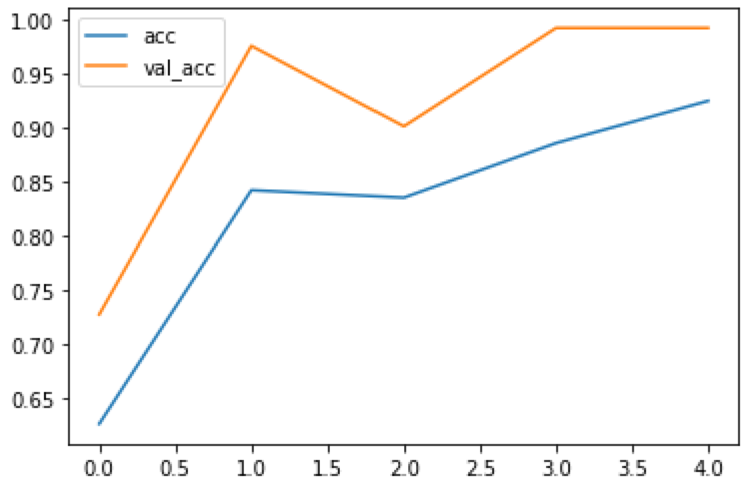
Out[ ]:



In [ ]:

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

Out[ ]:



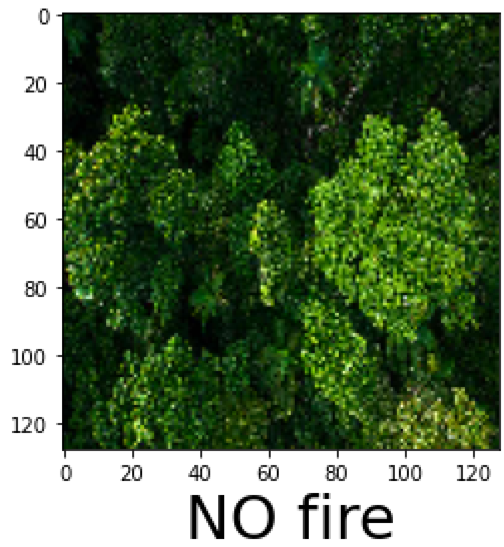
In [ ]:

```
def predictImage(filename):
  img1=image.load_img(filename,target_size=(128,128))
  plt.imshow(img1)
  y=image.img_to_array(img1)
  x=np.expand_dims(y,axis=0)
  val=model.predict(x)
  print(val)
  if val==0:
    plt.xlabel(" NO fire",fontsize=30)
  elif val==1:
    plt.xlabel("fire",fontsize=30)
```

In [ ]:

```
predictImage("/content/drive/MyDrive/forest-1.jpg")
```

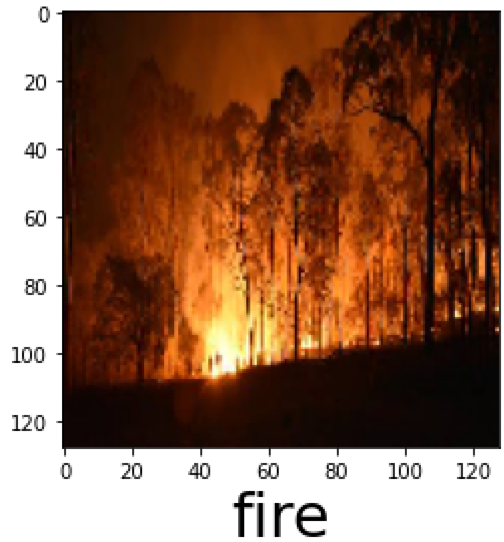
```
1/1 [==============================] - 0s 93ms/step
[[0.]]
```



In [ ]:

```
predictImage("/content/drive/MyDrive/forest-fire-1.jpg")
```

```
1/1 [==============================] - 0s 20ms/step
[[1.]]
```



Feature 2:
User alert message using Twilio regarding forest fire

pip install twilio
Successfully installed twilio

In [ ]:

pip install playsound

Successfully installed playsound-1.3.0

In [ ]:

pip install opencv-python

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (4.6.0.66)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python) (1.21.6)

In [ ]:

```
#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
```

In [ ]:

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

In [ ]:

```
video.isOpened()
```

Out[ ]:

True

In [ ]:

```
from tensorflow.keras.preprocessing import image
```

In [ ]:

```
from IPython.display import Audio
```

In [ ]:

```
while(video.isOpened()):
  success,frame=video.read()
  cv2.imwrite("image.jpg",frame)
  img=image.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)
  p=pred[0]
  print(pred)
    cv2.putText(frame,"predicted  class  =  ",(100,100),cv2.FONT_HERSHEY_SIMPLEX,  1,
(0,0,0), 1)
  if pred[0]==1:
    account_sid='AC2eb1ef0f60792aa19ad09be1f89a8dba'
    auth_token='a428f3fd3bd8ded0d44a6c4cbdd1945f'
    client=Client(account_sid,auth_token)
    message=client.messages \
    .create(
        body="Forest fire is detected ,stay alert",
        from_='+1 314 948 5657',
        to='+91 9344099941')
    print(message.sid)
    print('Fire detected')
    print('SMS sent')
    wn=Audio('/content/drive/MyDrive/alarm-sound.mp3',autoplay=True)
    display(wn)
    break
  else:
    print('No danger')
    break
  if cv2.waitKey(1) & 0xFF==ord('a'):
    break
video.release()
cv2.destroyAllWindows()
```

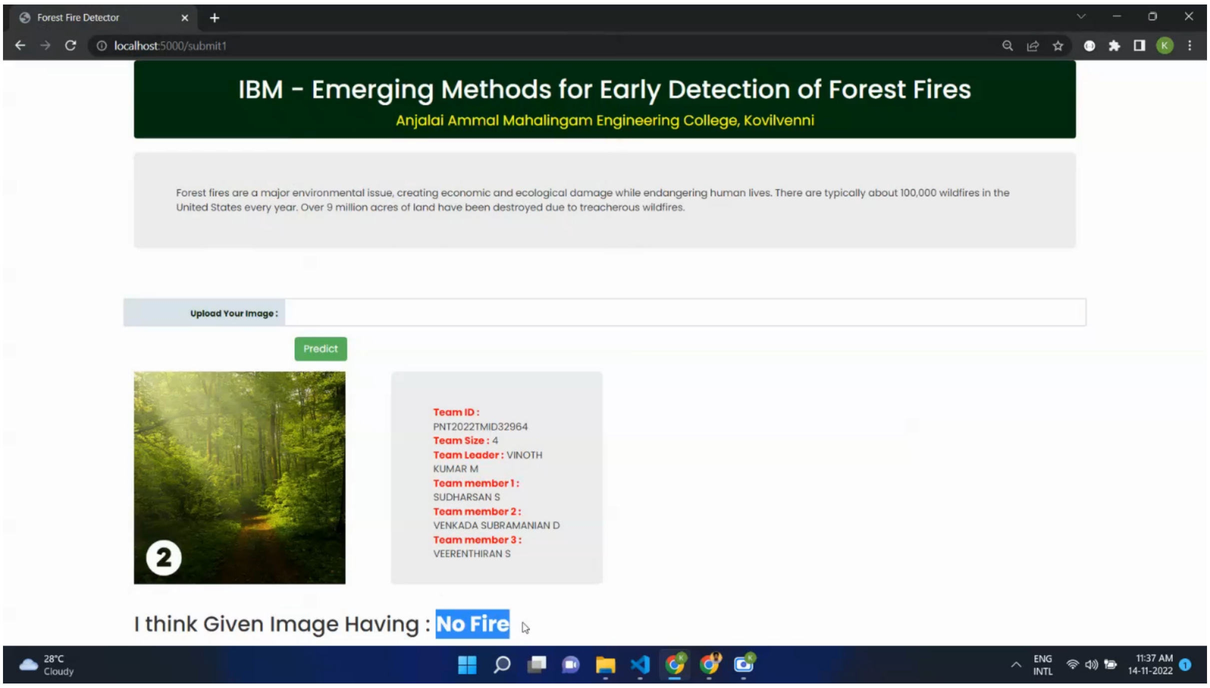
```
1/1 [==============================] - 0s 81ms/step
```

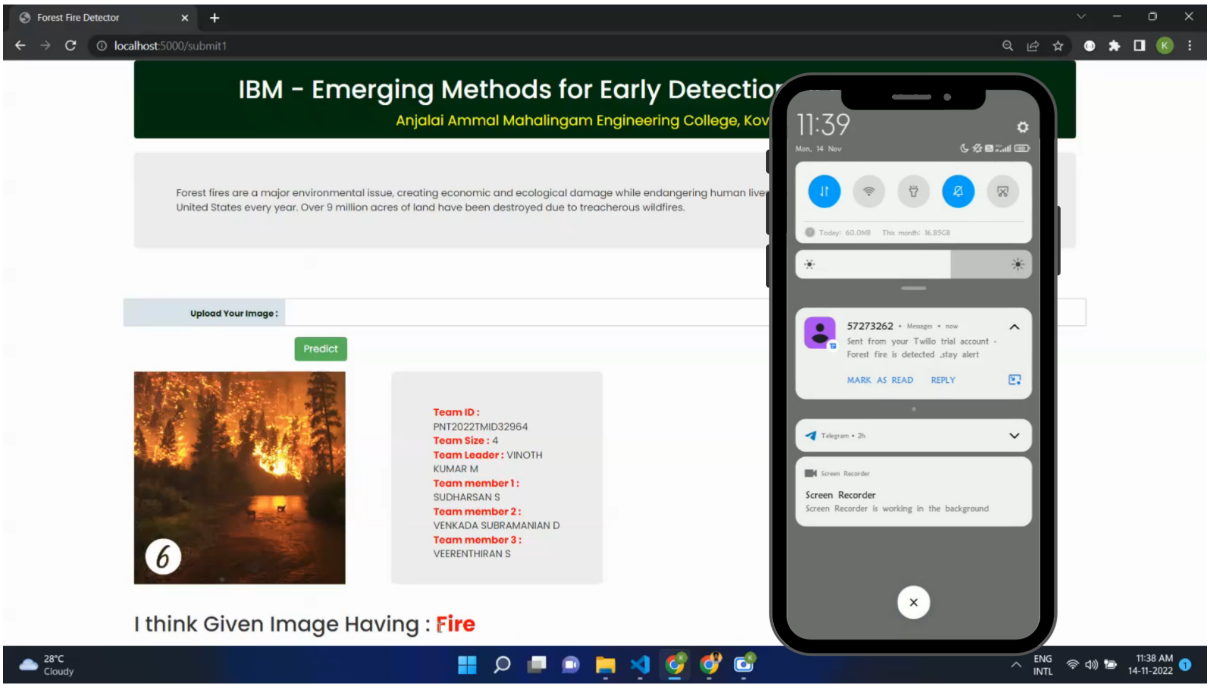
[[1.]]
SMc9da8fc345a2ce13c7e598ce055ad8fa
Fire detected

SMS sent

15. TESTING

Test Cases





User Acceptance Testing:

Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Test Case Analysis

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

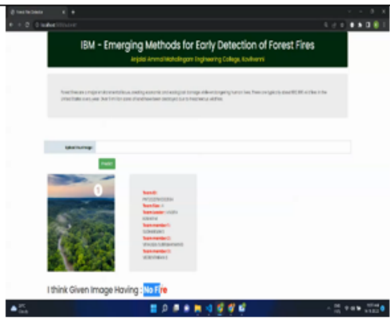
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

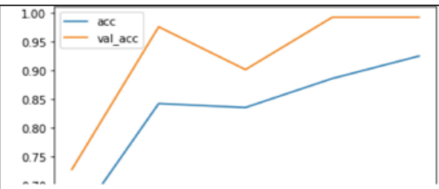
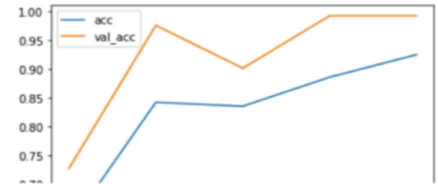
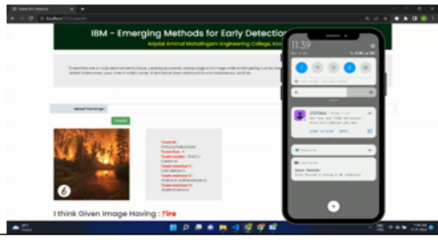
17.RESULTS

Performance Metrics:

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. 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 during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities.	

2.	Accuracy	Training Accuracy – 0.96 Validation Accuracy -0.96	 
3	Confidence Score (Only Yolo Projects)	Class Detected – No Fire Class Detected –Fire detected Confidence Score – 96/100	 

16. ADVANTAGES

The advantages of vision based fire detection techniques are listed here:

I. The fast response to fires.
II. The location of fire is sensed using this method not just the radiation,
III. The captured images can be analysed and it can be used for future purposes and storage,
IV. It can be used for outdoor places which cover a large area.

DISADVANTAGES

The disadvantages of vision based fire detection techniques are listed here:

I. Power consumption
II. Latency
III. Implementation costs.

CONCLUSION

The proposed Fire detection system uses Convolution Neural Network and Image Pre-processing techniques to detect Forest fire from the given Image and videos and send alert messages via Twilio API to users.
The proposed system is implemented with the real time datasets which signifies that the fire detection method is more appropriate for real-time unconstrained motion videos.

17. FUTURE SCOPE

- Fire detection in forest could also be possible if we used temperature sensors and humidity sensors along with the device which can also avoid wastage of valuable trees. Forest not only provides home to the large variety of flora and fauna, the animals but also the major producer of oxygen to the ecosystem.

- The sub server unit can be used between the transmitter unit and the main receiver unit which makes the whole procedure evenly proportional and take preventive measures to alert the forest officer.
- The system can be reformed with lower capacity components and OpenCV, making the system more efficient.

18. APPENDIX

Source Code:

GitHub:

https://github.com/IBM-EPBL/IBM-Project-45814-1660732535/tree/main/Final%20Deliverables

Project Demo Link:

https://www.canva.com/design/DAFR5rp6S-U/ImLHY7bulUCpNFJkj0B_lw/watch?utm_content=DAFR5rp6S-U&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink