EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRE

TEAM ID: PNT2022TMID13815

1. INTRODUCTION

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

1.1 Project Overview

The idea is to create aand develop a system that can identify the effects of the forest fire and it can analyse the forest fire by advanced AI techniques and CNN Algorithm then the Prediction model is Checked and then the model is connected with Twilio account credentials of the Developer consisting of phone numbers of the persons in the surroundings of the people in the area of easy forest fire zone then an security sound alert system is developed to make a alert sound which is downloaded from internet then the entire model is deployed to the IBM Cloud account that we have created.

1.2 Purpose

The forest fires destroys the wildlife habitat, damages the environment, affects the climate, spoils the biological properties of the soil, etc. So the forest fire detection is a major issue in the present decade. At the same time the forest fire have to be detected as fast as possible.

2. LITERATURE SURVEY

2.1 Existing problem

Forest fires have been and still are serious problem for theEuropean Union and for all other countries in Europe. In the year 2000, the EU has established the European Forest Fire Information system (EFFIS) [1], which will soon become part of the European Emergency Management Service, maintainedby the Copernicus Earth Observation Programme [2]. This system provides valuable near real-time and also historical data on the forest fires in Europe, the Middle East and North Africa. Currently EFFIS is being used and supported with data by 25 EU member states and by numerous other countries. According to the annual report of EFFIS for 2016 [3], more than 54 000 forest fires have occurred all around Europe and they have led to nearly 376 thousand hectares of burnt areas.If we compare these values to the average values from the EFFIS reports for the period 2006-2015, the number of forest fires have decreased by 13327 or by nearly 20%. This decrease can be explained with the more severe actions and sanctions towards the arsonists and with the introduction of more advanced technical solutions for early detection of the fires. Even though their number is decreasing, the forest fires continue to be extremely devastating events and they have destroyed just 27 thousand hectares (or 6.6 %) less than the average burnt areas for the period 2006-2015, according to [3]. Confirmation for this are the devastating forest fires form 2018, which took place in the Attica region of Greece and led to

more than 90 fatalities and to more than 200 injured people,as well as to the destruction to thousands of buildings [4]. Forest Fires can be divided into 4 categories in the forests of Hungary based on tree and other vegetation species: • underground burning, peat fire; • fire in undergrowth or dead fallen leaves; • fire in seedlings and saplings; • fire in trunks and shrouds.[5]

2.2 References

[1] Official webpage of the European Forest Fire Information System at:
http://effis.jrc.ec.europa.eu/
[2] Official webpage of the Copernicus Earth Observation Programme at:
http://www.copernicus.eu
[3] Forest Fires in Europe, Middle East and North Africa 2016, JRC Science for policy report, BN 978-92-79-71292-0, ISSN 1831-9424, doi:10.2760/17690, availabe at:
http://effis.jrc.ec.europa.eu/media/cms_page_media/40/Forest_fires_i
n_Europe_Middle_east_and_North_Africa_2016_final_pdf_JZU7He L.pdf
[4] The 2018 Attica wildfires Wikipedia webpage available at
https://en.wikipedia.org/wiki/2018_Attica_wildfires
[5] Rajmund Kuti,"Characterstic of forest fire and its impact on environment",(2016)

2.3 Problem Statement Definition

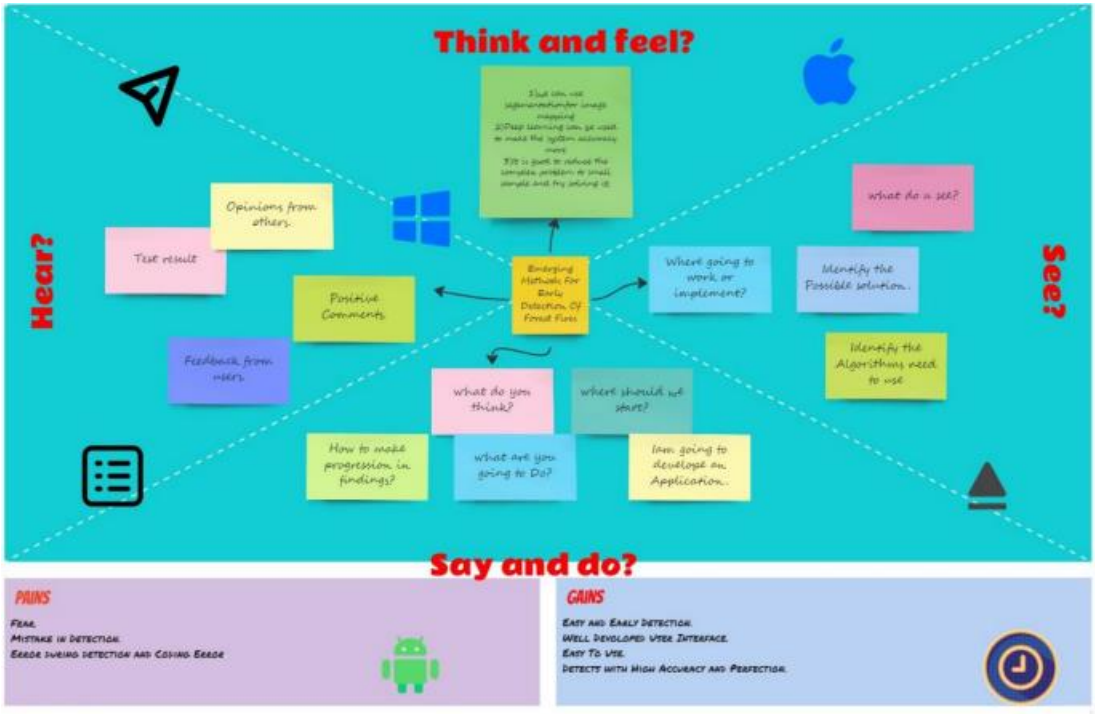
The user interacts with a web camera to read the video.
Once the input image from the video frame is sent to the model, if the fire is detected it is showcased on the console, and alerting sound will be generated and an alert message will be sent to the Authorities.

- Data Collection.
  - Collect the dataset or create the dataset.
- Image Preprocessing.
  - Import ImageDataGenerator Library.
  - Define the parameters /arguments for ImageDataGenerator class
  - Applying ImageDataGenerator on trainset and test set.
- Model Building
  - Import the model building Libraries
  - Initializing the model
  - Adding CNN Layers
  - Adding Hidden Layer
  - Adding Output Layer
  - Configure the Learning Process
  - Training and testing the model
  - Optimize the Model
  - Save the Model
- Video Streaming and alerting
  - OpenCV for video processing
  - Creating an account in Twilio service
  - Use Twilio API to send messages.

3. IDEATION AND PROPOSED SOLUTION

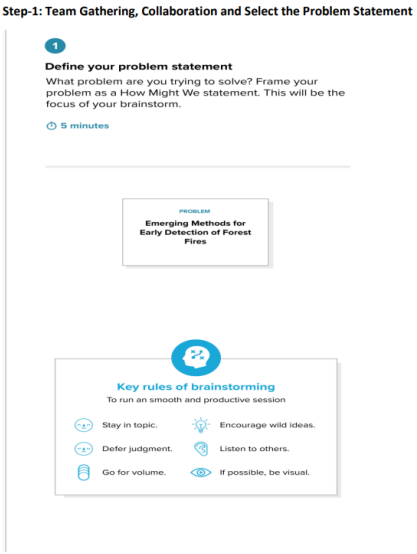
3.1 Empathy Map Canvas

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user's needs and pain points.

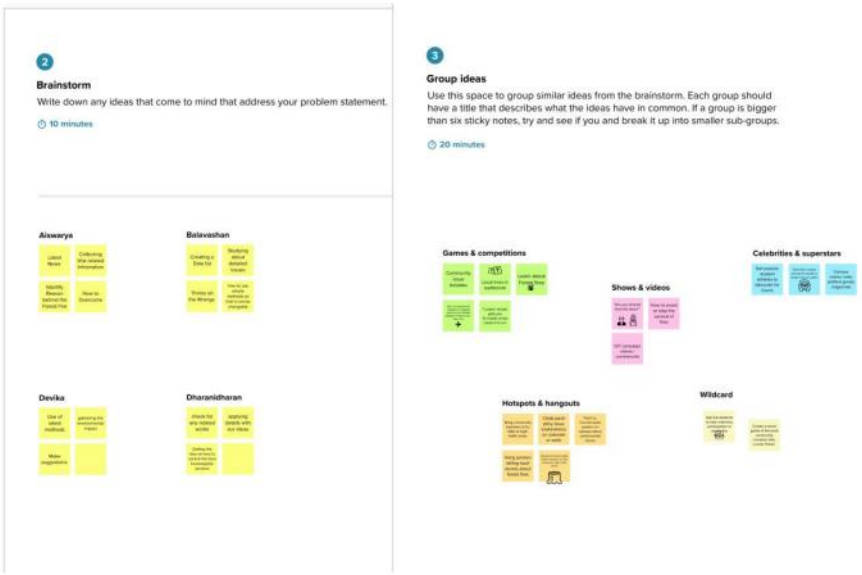


3.2 Ideation & Brainstorming

organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The aim is to Find the Emerging methods for Early Detection of Forest Fires (Using Artificial Intelligence).
2.	Idea / Solution description	**IDEA:** The idea is to create a system that can detect the forest fire and to give an alert message using twilio numbers and give an alert alarm sound using playsound. **SOLUTION:** The model using the pretrained image is constructed using Deep Learning technologies by CNN 2dconv networks to make detection more accurate and then this model is connected with the Open CV2 to make detection in video and images that was being captured then when the presence of fire an immediate alert message was sent to registered twilio account and following that an alert sound is played on the device.
3.	Novelty / Uniqueness	The system developed was very accurate as it can accurately detect and it is unique as idea was not yet developed.
4.	Social Impact / Customer Satisfaction	As forest fire was an important social impact that can cause many effects in living surroundings almost every living matters are affected by forest fire ,So our developed model can used for prior information about the forest fire to avoid it or to take safety prevention and to make alert of the peoples in the affected area.
5.	Business Model (Revenue Model)	This model is an economical model it can used in the place where the problems arises due to fire so that our model can detect perfectly to make prior warnings.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

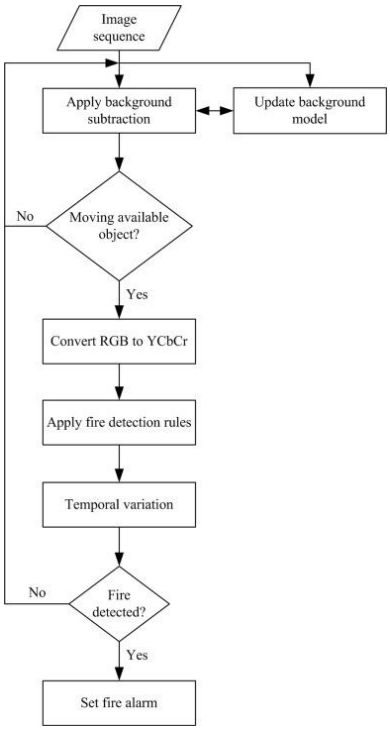
S.No	Component	Description	Technology
1.	Data Collection	Collecting and Analysing the raw Image Data.	Jupyter Notebook / python
2.	Image Preprocessing	Converting and correcting the image to make image quality and resolution high by rotation images in all possible directions and gaining knowledge.	Jupyter Notebook / python
3.	Trainset and Testset Image Data generation	Converting and correcting the image to make image quality and resolution high by rotation images in all possible directions and gaining knowledge for both trainset as well as testset data images.	Jupyter Notebook / python
4.	Model Building	Logic for Model by some Algorithms /Activation Functions.	Jupyter Notebook / python
5.	Saving the Model	Data Type, Configurations etc.	Python.
6.	Predictions	Make prediction of the trained model by checking its accuracy for the predictions.	DeepLearning / python
7.	Video Analysis	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Twilio Message service	Purpose of External API used in the application	IBM Weather API, etc.
9.	Alert Sound and Message	Sending Alert text message using registered twilio account and produce output sound alert alarm .	Twilio / PlaySound(Python)
10.	IBM Cloud	Create a IBM Cloud account to deploy the CNN model in cloud	Object Recognition Model / Deployment
11.	Train Model on Cloud	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration : and to train the deep learning model in IBM Cloud.	Local, IBM Cloud Account

4.2 Non-Functional requirements

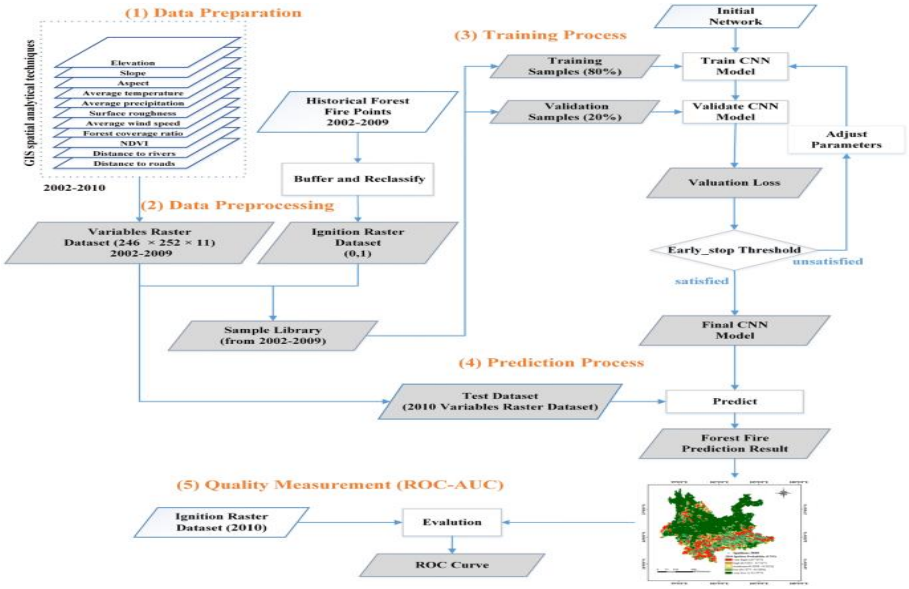
S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Jupyter Notebook,Twilio, Anaconda 3	Python
2.	Security Implementations	IBM Cloud	Bash /Python
3.	Scalable Architecture	Scalable for every Situation	Technology used was Java /Python
4.	Availability	Available on every windows versions.	Technology used - Python
5.	Performance	Detection Accuracy of 92%	Python

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Developer	Data Collection	USN-1	Collecting and Analysing the raw Image Data.	Through my Jupyter Notebook / google colab.	High	Sprint-1
Developer	Image Preprocessing	USN-1	Converting and correcting the image to make image quality and resolution high by rotation images in all possible directions and gaining knowledge.	Through my Jupyter Notebook / google colab.& click Run	High	Sprint-1
Developer	Trainset and Testset Image Data generation	USN-1	Converting and correcting the image to make image quality and resolution high by rotation images in all possible directions and gaining knowledge for test and train data.	Through my Jupyter Notebook / google colab.	Medium	Sprint-1
Developer	Model Building	USN-2	Logic for Model by some Algorithms /Activation Functions.	Through my Jupyter Notebook / google colab.	High	Sprint-2
	Saving the Model	USN-2	As a Developer saving the model developed for estimation of fire	Through my Jupyter Notebook / google colab.	High	Sprint-2
	Video Analysis	USN-3		Through my Dashboard	Medium	Sprint-3
Customer	Twilio Message service	USN-3		Twilio message service	Low	Sprint-3
Customer	Alert Sound and Message	USN-4	Sending Alert text message using registered twilio account and produce output sound alert alarm .	Playsound package	Low	Sprint-4
Administrator	Train Model on Cloud	USN-5	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration : and to train the deep learning model in IBM Cloud.	IBM Cloud deployment service	Medium	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-2	Saving the Model	USN-2	As a Developer saving the model developed for estimation of fire	10	High	Aiswarya S Balavashan RS Devika S Dharanidharan SS
Sprint-3	Video Analysis	USN-3		10	Medium	Aiswarya S Balavashan RS Devika S Dharanidharan SS
Sprint-3	Twilio Message service	USN-3		10	Low	Aiswarya S Balavashan RS Devika S Dharanidharan SS
Sprint-4	Alert Sound and Message	USN-4	Sending Alert text message using registered twilio account and produce output sound alert alarm .	10	Low	Aiswarya S Balavashan RS
Sprint-4	Train Model on Cloud	USN-5	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration : and to train the deep learning model in IBM Cloud.	10	Medium	Devika S Dharanidharan SS

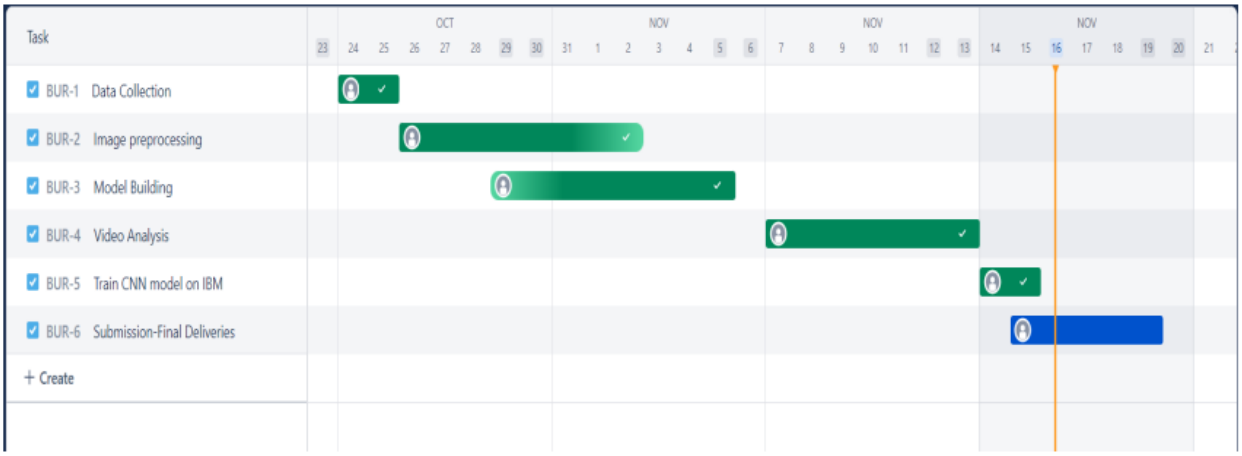
6.2 Sprint Delivery Schedule

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

The following table shows the sprint works assigned to the members along with the priority and story points assigned with the functional requirements with regards to user story.

6.3 Reports from JIRA

Burndown Chart:



7. CODING & SOLUTION

7.1 Feature 1

In Feature 1 module we have made data collection and Image preprocessing for and Model training.

importing Required Libraries:

```
import keras
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import numpy as np
batch_size = 32
```

image resizing and preprocessing :

```
train_datagen = ImageDataGenerator(
```

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

val_datagen = ImageDataGenerator(
    rescale=1./255
)
train_generator = train_datagen.flow_from_directory(
    'train_set/',
    target_size=(150, 150),
    batch_size=batch_size,
    class_mode='binary'
)

val_generator = val_datagen.flow_from_directory(
    'test_set/',
    target_size=(150, 150),
    batch_size=batch_size,
    class_mode='binary'
)
```

Creating the sequential model :

```
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Activation
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers import Dense

model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(150,150,3)))  #Convolutional 2D Layer
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))  # MaxPooling Layer
model.add(Flatten())        #Flatten Layer to make a array
model.add(Dense(150))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
model.compile(
    loss='binary_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

Model summary :

model.summary()

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
activation (Activation)      (None, 148, 148, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 175232)            0
_________________________________________________________________
dense (Dense)                (None, 150)               26284950
_________________________________________________________________
activation_1 (Activation)    (None, 150)               0
_________________________________________________________________
dropout (Dropout)            (None, 150)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 151
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 26,285,997
Trainable params: 26,285,997
Non-trainable params: 0
_________________________________________________________________
```

7.2 Feature 2

```
from keras.models import load_model
from keras.preprocessing import image
import numpy as np
import cv2
from PIL import Image, ImageOps
model=load_model("forest1.h5")
from twilio.rest import Client
from playsound import playsound
model=load_model('forest1.h5')
video=cv2.VideoCapture(0)
name=['forest','with fire']
account_sid='ACca0e8bb11699d2957d67c979ca84b68a'
auth_token='bcb5f3850ef4b7ed263f60efc9acecdb'
client =Client(account_sid,auth_token)
message=client.messages \
.create(
body='-------Forest Fire is detected,Stay Alert !!!--------',
   from_='+19457581434',to='+919943435141')
```

```
print(message.sid), print("Alert Message sent")
```

```
SMb8a51eaeb987fbc8d5eced2dab56300a
Alert Message sent
```

8. TESTING

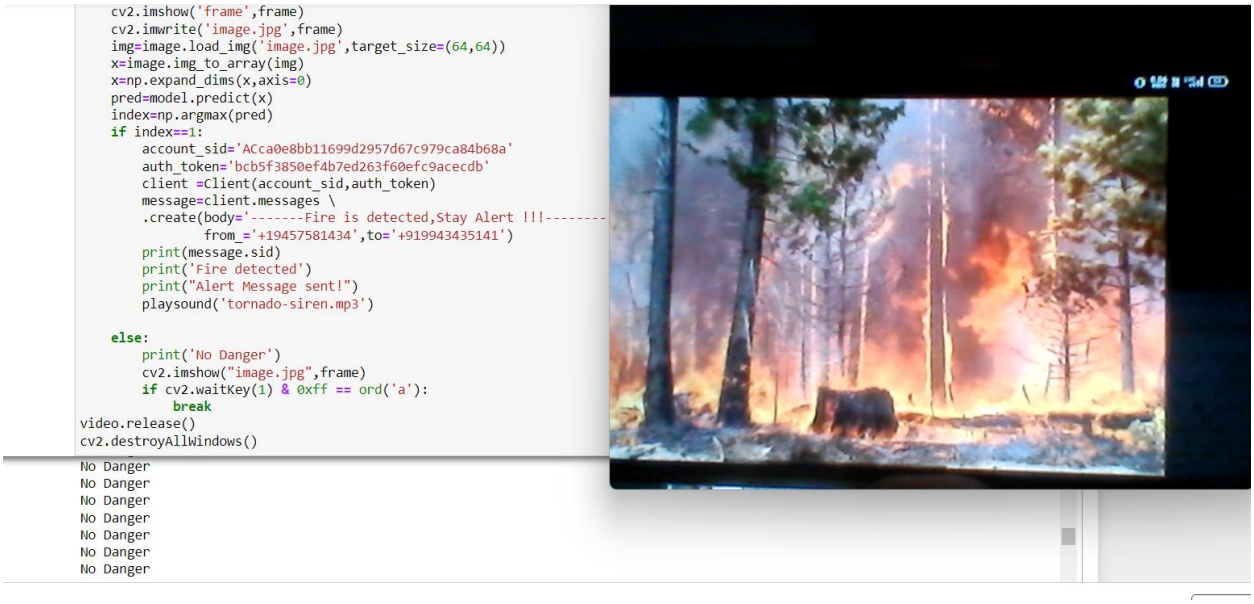
8.1 Test Cases & User Acceptance Testing

Testing with input video recording from user end:

```
import cv2
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
from twilio.rest import Client
from playsound import playsound
model=load_model('forest1.h5')
video=cv2.VideoCapture(0)
name=['forest','with fire']
while(True):
    ret,frame=video.read()
    cv2.imshow('frame',frame)
    cv2.imwrite('image.jpg',frame)
    img=image.load_img('image.jpg',target_size=(64,64))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    pred=model.predict(x)
    index=np.argmax(pred)
    if index==0:
        account_sid='ACca0e8bb11699d2957d67c979ca84b68a'
        auth_token='bcb5f3850ef4b7ed263f60efc9acecdb'
        client =Client(account_sid,auth_token)
        message=client.messages \
        .create(body='-------Fire is detected,Stay Alert !!!--------',
                from_='+19457581434',to='+919943435141')
        print(message.sid)
        print('Fire detected')
        print("Alert Message sent!")
        playsound('tornado-siren.mp3')

    else:
        print('No Danger')
        cv2.imshow("image.jpg",frame)
        if cv2.waitKey(2) & 0xff == ord('a'):
            break
video.release()
cv2.destroyAllWindows()
```

output for user input video stream



9.RESULTS

9.1 Performance Metrics

```
loss: 0.3438 - accuracy: 0.8483 - val_loss: 0.2485 - val_accuracy: 0.958

loss: 0.3816 - accuracy: 0.8483 - val_loss: 0.2569 - val_accuracy: 0.958

loss: 0.4068 - accuracy: 0.8391 - val_loss: 0.2547 - val_accuracy: 0.958

loss: 0.3312 - accuracy: 0.8437 - val_loss: 0.2601 - val_accuracy: 0.950

loss: 0.5621 - accuracy: 0.8368 - val_loss: 0.2679 - val_accuracy: 0.958
```

10. ADVANTAGES & DISADVANTAGES

Advantages

- Easily detect and Estimate the Forest Fire.
- Most Accurate
- Flexible Model which can give maximized outcome
- No Specific Requirements needed to implement the model

Disadvanatges

- Training model is time consuming process.
- Error in Cv can cause damage to camera
- Access of camera are prohibited due to personal issues

11.CONCLUSION

Thus we have constructed a model that can can identify the effects of the forest fire and it can analyse the forest fire by advanced AI techniques and CNN Algorithm then the Prediction model is Checked and then the model is connected with Twilio account credentials of the Developer consisting of phone numbers of the persons in the surroundings of the people in the area of easy forest fire zone then an security sound alert system is developed to make a alert sound which is downloaded from internet then the entire model is deployed to the IBM Cloud account that we have created was made with the studies we have done.

12.FUTURE SCOPES

- It can be developed as a Web or Android Application.
- In future Alternate Advanced technologies can be Implemented.
- The Identification and tracking system can be implemented if possible.

14. APPENDIX

Source Code

SourceCode:https://github.com/IBM-EPBL/IBM-Project-33821-1660227659/blob/main/Final%20Deliverables/EntireModel.ipynb

GitHub & Project Demo Link

Github: https://github.com/IBM-EPBL/IBM-Project-33821-1660227659

Demo Link: https://drive.google.com/drive/u/1/folders/1xo6bxRthoGpOdThyTDgzzR8lshLESl4m