IBM - NALAIYA THIRAN
PROJECT DOCUMENTATION
on
EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES.

Submitted by

SHESHANK C	**AC19UCS106**
SHIKSHA N	**AC19UCS107**
SURYA K S	**AC19UCS122**
SWATHI SHREE K V	**AC19UCS124**

in partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING

REPO ID : IBM-Project-12494-1659452250

TEAM ID : PNT2022TMID08041

Faculty Mentor : Mrs. J Reshma Farhin

Industrial Mentor : Shanthi

CONTENTS

CHAPTER - 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Detecting fire at an early stage before they turns into a catastrophic event is crucial to prevent fire disastrous and save people's lives and properties. Forest fires have become a major threat around the world, causing negative impacts on wildlife and their habitats ,not only this, they also disturb the natural functioning of the environment and human beings. It causes deterioration of the air quality which when inhaled becomes the principal public health threat. They would emit CO_2 and other greenhouse gases, which is also a cause for ozone hole. They also become a source for air pollution,water pollution and land pollution. This has an impact not only in the present but also continues to increase in the future. An average of approximately 70,000 forest fires are happening per year all over the world which is being a notable threat to the environment.

1.2 PURPOSE.

This project aims to provide an effective alert system to the forest fire officers in case of a fire being detected at the most earliest possible time ,so as to save the wildlife & its habitats and also to prevent the environment from being polluted. The best part is the accuracy of prediction is very high when comparing with all the existing projects.

CHAPTER - 2

LITERATURE SURVEY

2.1 Existing problem

- 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. In image processing the inputs for the fire detection may be an image or a video but the input as a video is quite a complex process but provides good results. The techniques such as Wavelet decomposition, spatial and temporal analysis, Gaussian Mixture Model, Multi-Feature fusion detect fire in an accurate manner.
- Both a full image and fine grained patch fire classifier in a joined deep convolutional neural network (CNN). The fire detection is operated in a cascaded fashion, ie the full image is first tested by the global image-level classifier, if fire is detected, the fine grained patch classifier is followed to detect the precise location of fire patches.To facilitate the evaluation of various fire detectors in the community, we build a fire detection benchmark. According to our best knowledge, this is the first one with patch-level annotations.
- It is a well known fact that the sooner a wildfire is detected, the quicker it can be put out, which highlights the importance of early detection. By scanning the landscape using regular cameras and Deep Artificial Neural Networks, Bee2Fire searches for smoke columns above the horizon with an image classification approach. After these networks were trained, the system was deployed in the field, obtaining a sensitivity score between 74% and 93%, a specificity of more than 99% and a precision of around 82%.

2.2 REFERENCES.

A Survey on Forest Fire Detection:
https://www.researchgate.net/publication/305190416

Deep Convolutional Neural Networks for Forest Fire Detection:
https://www.researchgate.net/publication/315562504_Deep_Convolutional_Neural_Networks_for_Forest_Fire_Detection

A Deep Learning Powered Forest Fire Detection System:
https://www.scitepress.org/Papers/2020/89661

2.3 PROBLEM STATEMENT DEFINITION.

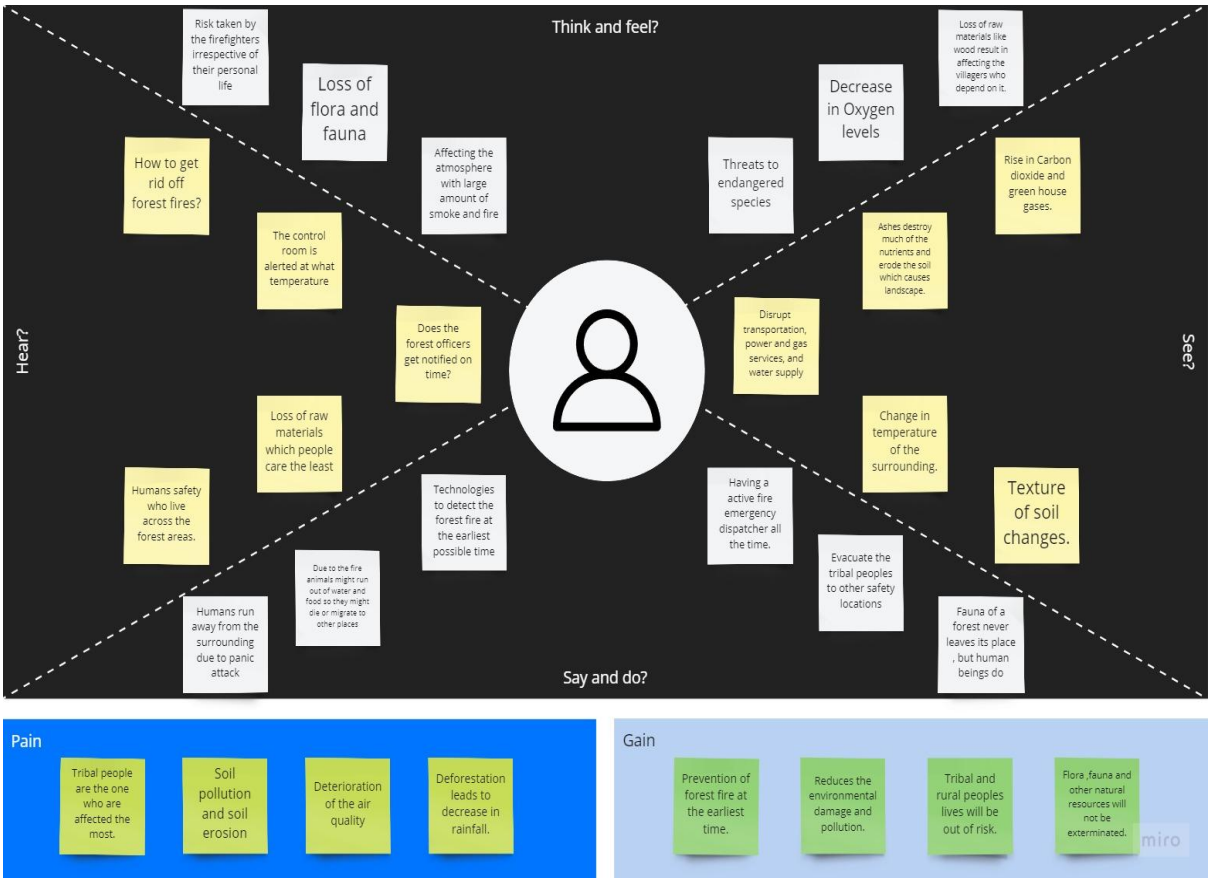
Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. It causes deterioration of the air quality which when inhaled becomes the principal public health threat. They would emit CO_2 and other greenhouse gases which are also very harmful to every living things. To detect forest fires there are many algorithms and techniques which include sensors, carbon dioxide level detection in the air,unmanned aerial vehicles, image acquisition, B&W spatio temporal algorithm etc.. . But these don't provide good accuracy and there is a lack of efficiency. To achieve the required accuracy we have now employed digital camera technology and video processing techniques, since there is a major trend to replace conventional fire detection methods with computer vision based systems. To this feature extraction is implemented using deep learning techniques and the forest fire is predicted with an efficient alert system. The main objective of this is to predict the forest fire at the earliest possible time with good accuracy and to alert the forest fire officers.

CHAPTER-3
IDEATION & 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 that helps teams to 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.

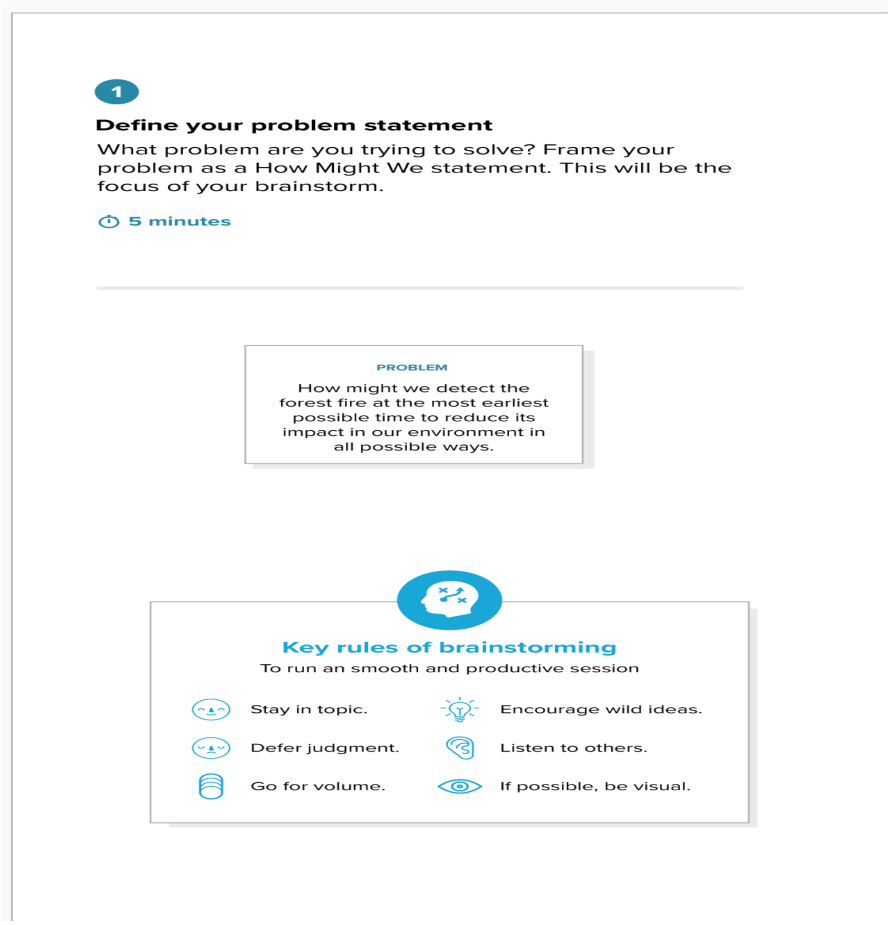


3.2 IDEATION & BRAINSTORMING.

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritising 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 their imagination and start shaping concepts even if you're not sitting in the same room.

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

In this step team members gather and provide their ideas and collaborate those ideas and select their problem statement.The ideas should be relevant to their problem statement.



Step-2: Brainstorm, Idea Listing and Grouping.

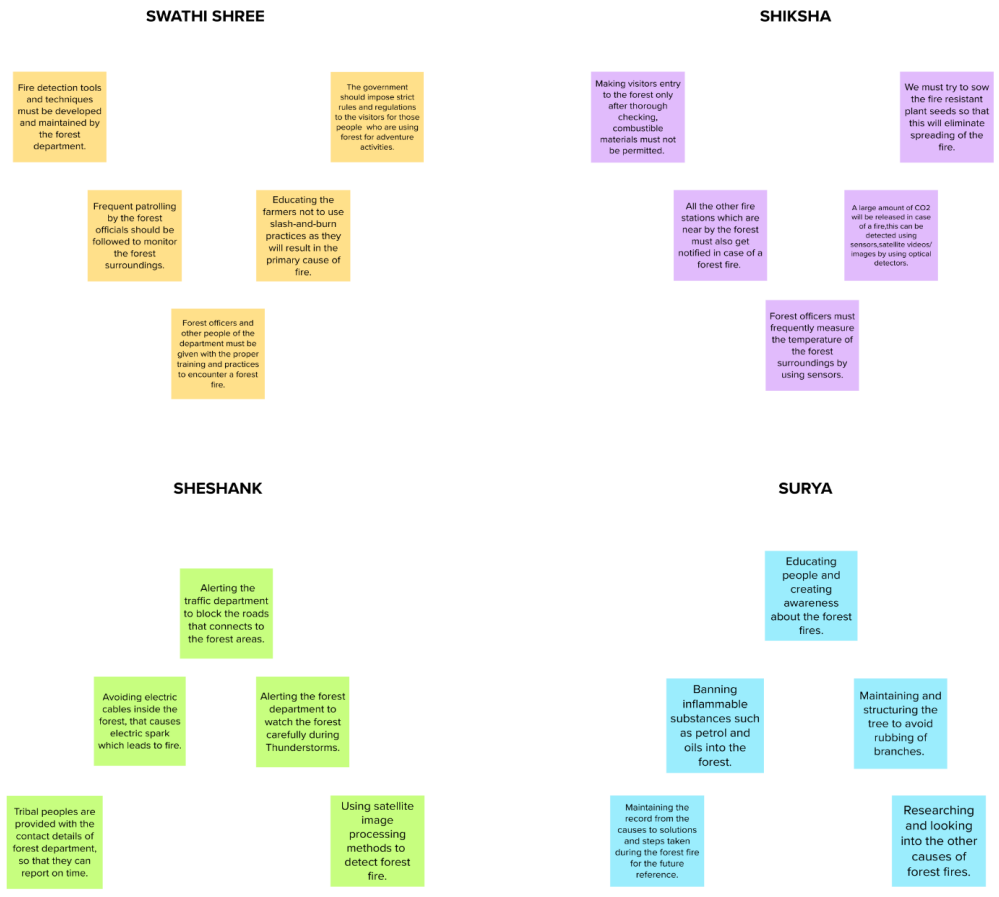
In this step team members gather and provide their ideas and collaborate those ideas and select their problem statement.The ideas should be relevant to their problem statement.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes



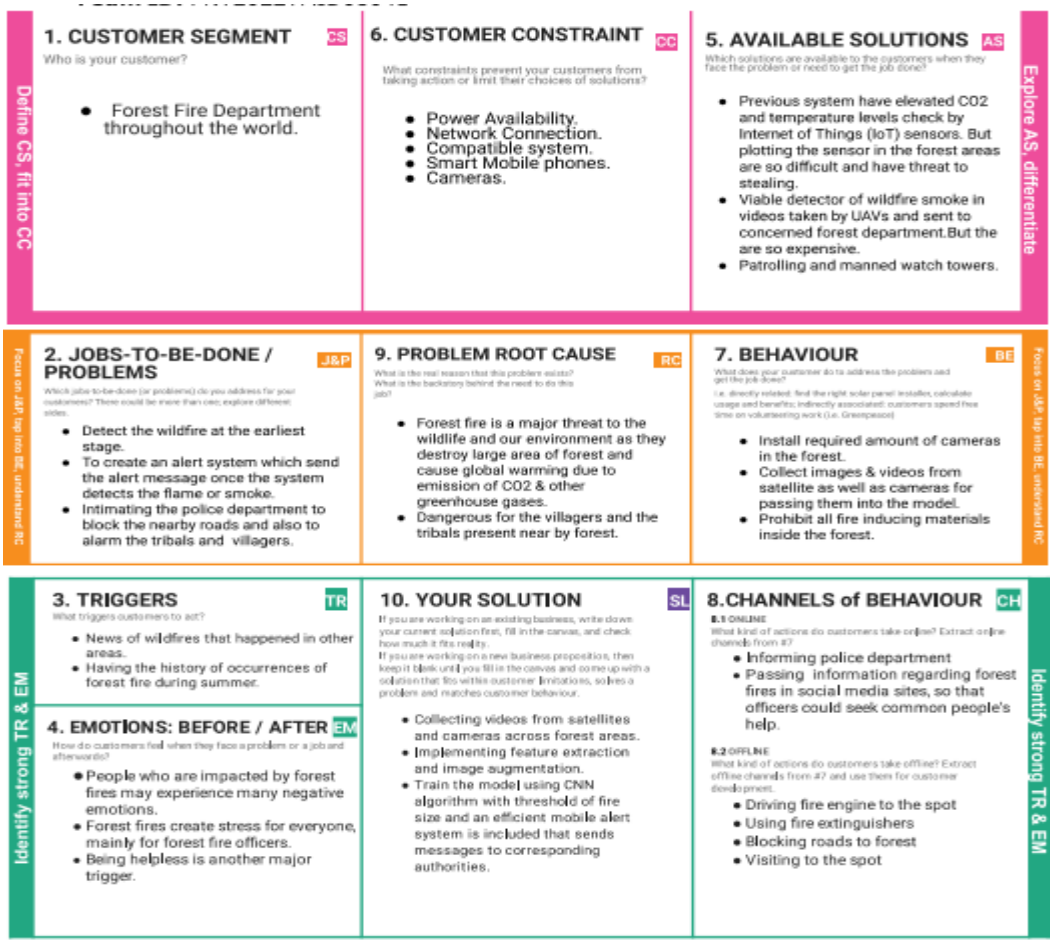
Step-3: Idea Prioritization

As mentioned, idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated into everyday ways of working.

3.3 PROPOSED SOLUTION.

- Problem Statement is to detect the forest fire at the most earliest possible time to reduce its impact on our environment in all possible ways. This is done using CNN algorithm where feature extraction is the main idea.
- Solution description -We implement preprocessing steps to eliminate the noises in images. And also implements feature extraction to extract the color features and segment of the fire regions.
- Novelty / Uniqueness-We classify the pixels using CNN algorithm with an efficient mobile alert system that sends messages to corresponding authorities.
- Social Impact / Customer Satisfaction- Saves the environment and its related resources from great loss. Prevent damaging of flora,fauna and some of the important endangered species. Large amounts of CO2 emissions are avoided.

3.4 PROBLEM SOLUTION FIT.



CHAPTER - 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS.

- **FR - 1**

  Functional Requirement (Epic) - Collecting Dataset

  Sub Requirement (Story/Sub-Task) - Dataset from different platforms and realtime data.

- **FR - 2**

  Functional Requirement (Epic) - Data Preprocessing

  Sub Requirement (Story/Sub-Task) - The collected data are imported and checked for the augmentation and splitting of the dataset .

- **FR - 3**

  Functional Requirement (Epic) - Model Building

  Sub Requirement (Story/Sub-Task) - Adding of different layers to the CNN model like convolutional layer,pooling layer and dense layer.

- **FR - 4**

  Functional Requirement (Epic) - Alarm System.

  Sub Requirement (Story/Sub-Task) - This system is built in the way that on detecting the fire without any delay the alarm messages are sent to the forest department.

- **FR - 5**

  Functional Requirement (Epic) - Tracking report.

  Sub Requirement (Story/Sub-Task) - We keep a track on the events that occur in the forest so that in the near future we can avoid unfavourable situations.

- **FR - 6**

  Functional Requirement (Epic) - Storage.

  Sub Requirement (Story/Sub-Task) - Store the information of previously occurred forest fires and the measures taken at that time which will be helpful for future references to take actions faster. These data are accessible by only the registered users.

4.2 NON-FUNCTIONAL REQUIREMENTS.

- **NFR-1**

  Non-Functional - Usability

  Requirement Description - The system is built to detect forest fire through images and videos uploaded, it is made more user-friendly and easy enough that even someone without technological expertise can use it.

- **NFR-2**

  Non-Functional - Security.

  Requirement Description - This system keeps a high level of security. It does not accept corrupted media into the system. Not everyone has access to the system.

- **NFR-3**

  Non-Functional - Reliability.

  Requirement Description - Possibility of a false alarm is very low as the system is trained with all stages of forest fires and no forest fire conditions.

- **NFR-4**

  Non-Functional - Performance.

  Requirement Description - System shows high accuracy in detecting the fire and sends the alert message without any delay or lag.

- **NFR-5**

  Non-Functional - Availability.

  Requirement Description - The system works 24/7 everyday and can even analyse the live videos. Incase of fire they alarm us by sending alert messages immediately.

- **NFR-6**

  Non-Functional - Scalability.

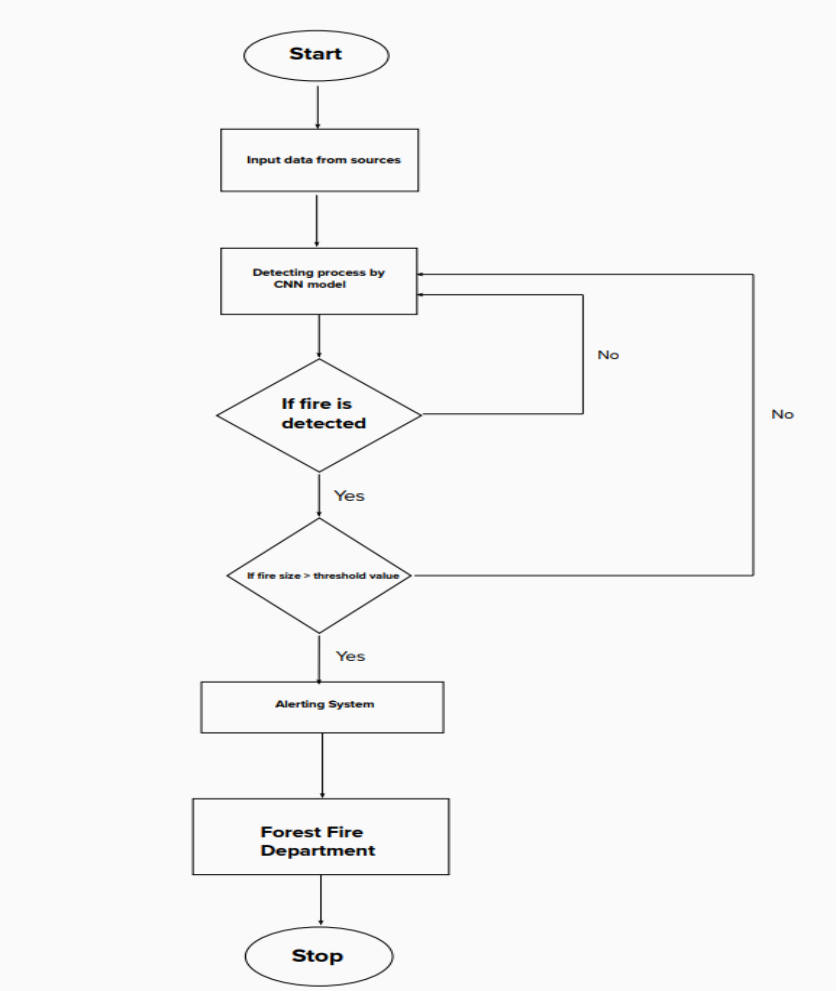
  Requirement Description - Faster and efficient detection of fire even if the data are larger scale.

CHAPTER - 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS.

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.



DFD Level 0 (Industry Standard)

Continous Monitoring System

Training the CNN Model

CNN deployment on IBM

Trained CNN Model

No

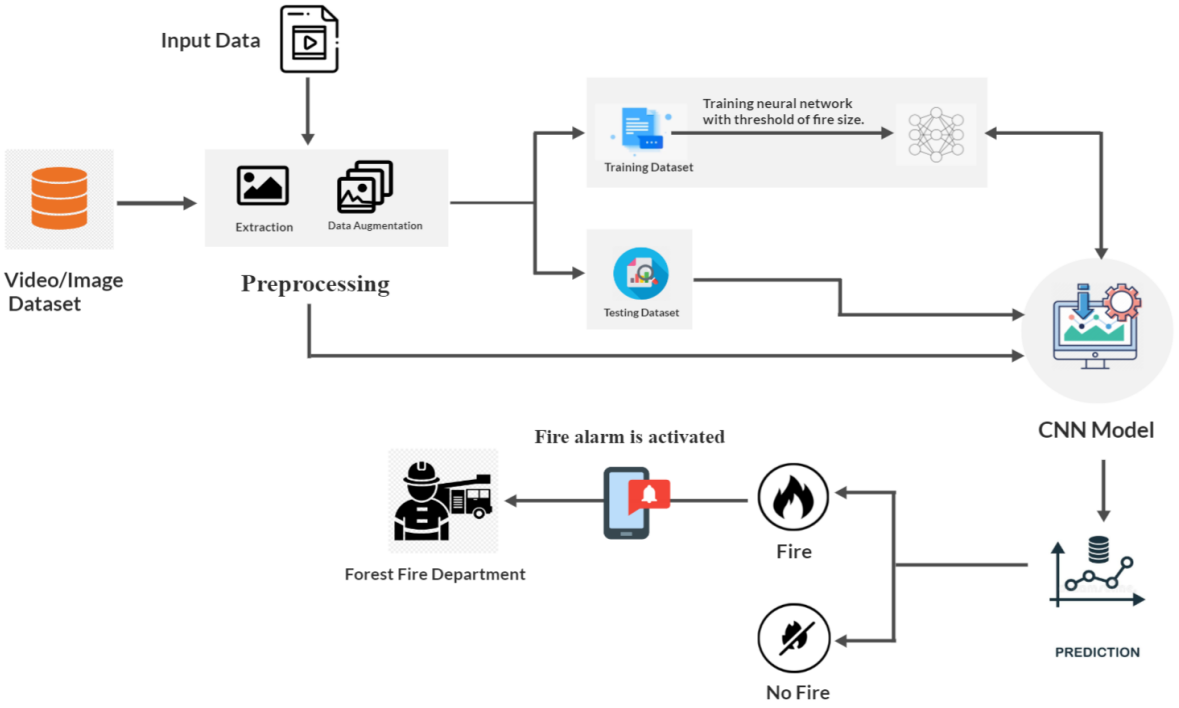
Fire Detection

Yes

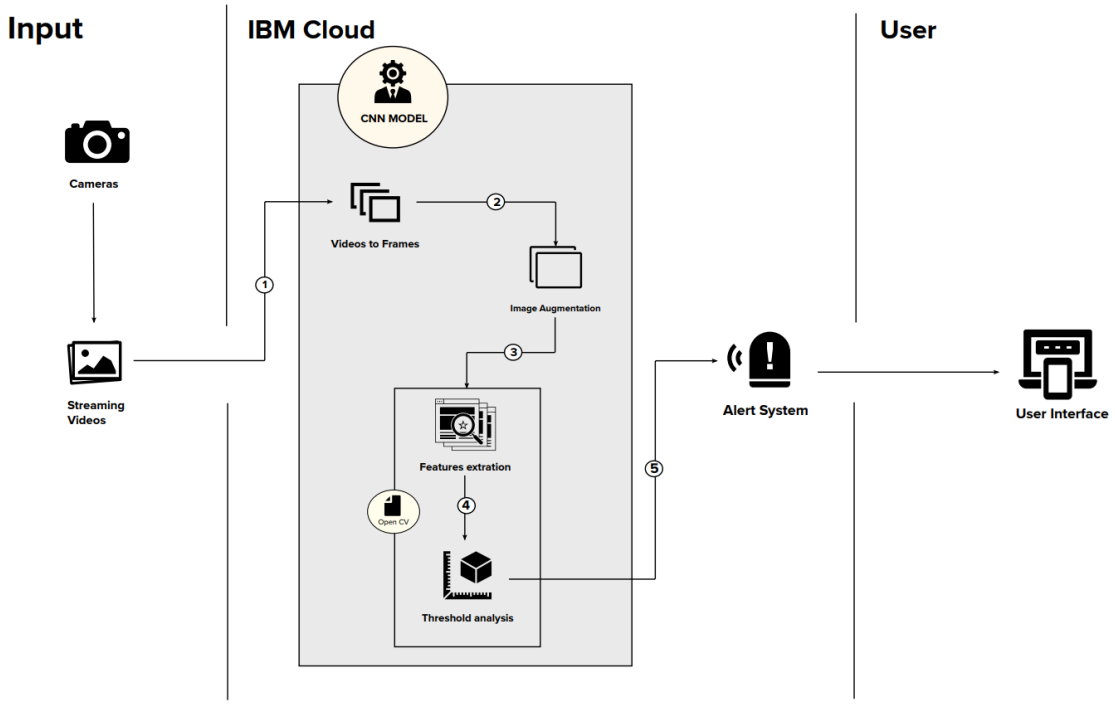
Alert the customers by sending messages.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE.

Solution architecture:



Technical architecture:



Components & Technologies:

1. **Component-** Input sources.

   Description- Videos and images.

   Technology- Surveillance cameras.

2. **Component-** Preprocessing.

   Description- Converting videos to frames,performing image augmentation.

   Technology- Python

3. **Component-** Model Creation.

   Description- Performing feature extraction and threshold analysis.

   Technology- Python,OpenCV.

4. **Component-** Database.

   Description- Data Type, Configurations etc.

   Technology- MySQL.

5. **Component-** Cloud Database

   Description-Database Service on Cloud

   Technology- IBM DB2,AWS etc.

6. **Component-** File Storage.

   Description- File storage requirements.

   Technology- IBM Block Storage.

7. **Component-** External API-1

   Description-Purpose of External API used in the application.

   Technology- IBM Weather API, etc.

8. **Component-** Deep Learning Model.

   Description- Purpose of deep Learning Model.

   Technology- Convolution neural network model.

9. **Component-** Infrastructure (Server / Cloud)

   Description- Application Deployment on Local System / Cloud.

   Technology- Local, Cloud Foundry.

Application Characteristics:

1. **Characteristics-**Open-Source Frameworks.

   Description-List the open-source frameworks used.

   Technology-Google Colab,Tensorflow,Fire-Net.

2. **Characteristics-**Security Implementations.

   Description-List all the security / access controls implemented, use of firewalls etc.

   Technology-e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.

3. **Characteristics-**Scalable Architecture.

   Description-Justify the scalability of architecture.

   Technology-Python,Anaconda.

4. **Characteristics-** Availability.

   Description- Justify the availability of application.

   Technology-IBM load balancer.

5. **Characteristics-**Performance.

   Description-Design consideration for the performance of the application.

   Technology-We use CDN for fast processing of videos .

5.3 USER STORIES.

Sprint-01:

Usn-01 -Collecting the dataset.

Usn-02 -Performing data augmentation using ImageDataGenarator.

Usn-03 -Splitting the dataset into training data and testing data.

Sprint-02:

Usn-04 - Model is trained with all different images of forest fires and in different dimensions.By training, the model could learn how the forest fire looks. It has medium priority.

Usn-05 -The model is implemented to check its working and accuracy. Accuracy of the algorithm is calculated with testing data. It has medium priority.

Usn-06 -Once the threshold an alarm will be sent to the respective user (forest department officer) .It has high priority.

Sprint-03:

Usn-07 -Detects the forest fire with the trained model given by the cameras as a stream of images or videos.It has high priority.

Sprint-04:

Usn-08 -On accurate detection of forest fire in early time the alarm messages are sent and necessary action to extinguish the fire is taken.It has high priority.

CHAPTER - 6
PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION.

Sprint-01:

Usn-01 -As a user, I can register for the application by using government certified IDs.I can access the account. It has high priority.

Team member

- **SHIKSHA N**

Usn-02 -As a user, I will authenticate using ID credentials. It has high priority.

Team members

- **SHESHANK C**
- **SURYA K S**

Usn-03 -With the credentials the user can login easily. It has low priority.

Team member

- **SWATHI SHREE K V**

Sprint-02:

Usn-04 -Model is trained with all different images of forest fires and in different dimensions.By training, the model could learn how the forest fire looks. It has medium priority.

Team members

- **SHESHANK C**
- **SHIKSHA N**

Usn-05 -The model is implemented to check its working and accuracy. Accuracy of the algorithm is calculated with testing data. It has medium priority.

Team members

- **SURYA K S**
- **SWATHI SHREE K V**

Usn-06-Once the cross the threshold an alarm will be send to the respective user (forest department officer) .It has high priority.

Team member

- **SHESHANK C**

Sprint-03:

Usn-07 -Detects the forest fire with the trained model given by the cameras as a stream of images or videos.It has high priority.

Team members

- **SHIKSHA N**
- **SWATHI SHREE K V**

Sprint-04:

Usn-08 -On accurate detection of forest fire in early time the alarm messages are sent and necessary action to extinguish the fire is taken.It has high priority.

Team members

- **SHESHANK C**
- **SWATHI SHREE K V**

6.2 SPRINT DELIVERY SCHEDULE.

Sprint-1

- **Total Story Points -** 20
- **Duration -** 7 Days.
- **Sprint Start Date -** 24 Oct 2022.
- **Sprint End Date (Planned) -** 29 Oct 2022.
- **Story Points Completed (as on Planned End Date) -** 20
- **Sprint Release Date (Actual) -** 30 Oct 2022.

Sprint-2

- **Total Story Points -** 20
- **Duration -** 7 Days.
- **Sprint Start Date -** 31 Oct 2022.
- **Sprint End Date (Planned) -** 05 Nov 2022.
- **Story Points Completed (as on Planned End Date) -** 20
- **Sprint Release Date (Actual) -** 06 Nov 2022.

Sprint-3

- **Total Story Points -**20
- **Duration -** 7 Days.
- **Sprint Start Date -** 07 Nov 2022.
- **Sprint End Date (Planned) -** 12 Nov 2022.
- **Story Points Completed (as on Planned End Date) -** 20
- **Sprint Release Date (Actual) -** 13 Nov 2022.

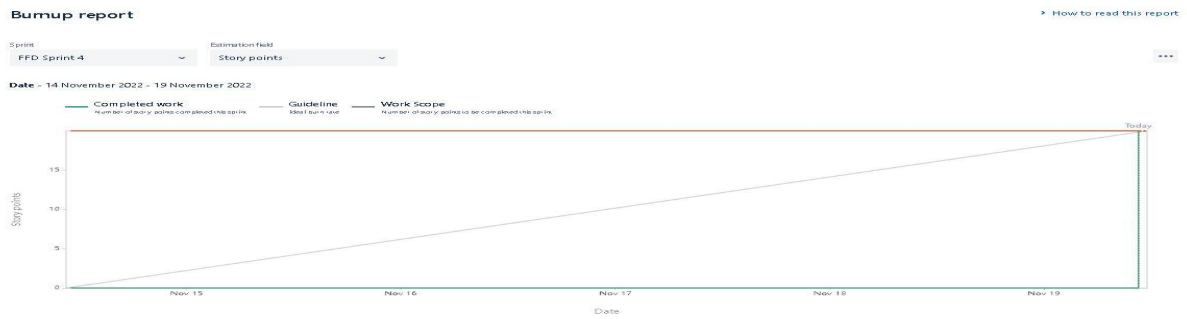
Sprint-4

- **Total Story Points -** 20
- **Duration -** 7 Days.
- **Sprint Start Date -** 14 Nov 2022.
- **Sprint End Date (Planned) -** 19 Nov 2022.
- **Story Points Completed (as on Planned End Date) -** 20
- **Sprint Release Date (Actual) -** 20 Nov 2022.

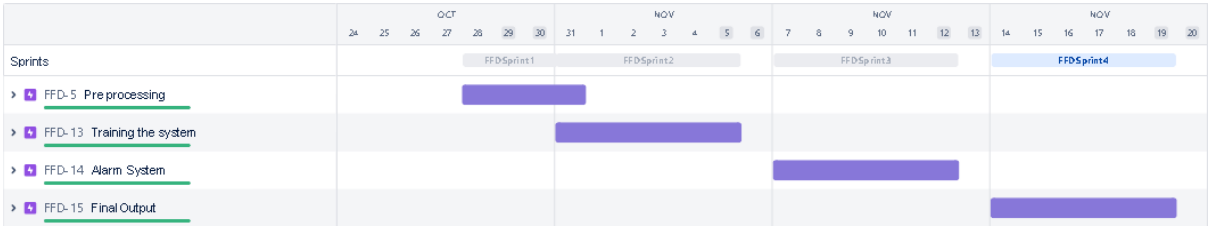
6.3 REPORTS FROM JIRA

Burndown chart report :

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



Road map



CHAPTER - 7
CODING & SOLUTIONING

7.1 Feature 1

Data Preprocessing:

Pre-processing is the transformations applied to our data before feeding it to the deep learning model. Data Preprocessing is a technique that is used to convert the raw data into a clean data set.The keras library package includes the data preprocessing modules,where the ImageDataGenarator is used.Keras image data generator class is used to carry out data augmentation where we aim to gain the overall increment in the generalization of the model. Operations such as rotations, translations, shearin, scale changes, and horizontal flips are carried out randomly in data augmentation using an image data generator.

```
[1] #Importing the ImageDataGenerator class
    import keras
    from keras.preprocessing.image import ImageDataGenerator
```

```
[2] #Define the parameters/arguments for ImageDataGenerator class
    train_data = ImageDataGenerator(rescale= 1./255,shear_range= 0.2,rotation_range= 180,horizontal_flip = True,vertical_flip = True,zoom_range = 0.
```

```
[3] test_data = ImageDataGenerator(rescale= 1./255)
```

```
[4] #Applying ImageDataGenerator functionality to trainset
    x_train = train_data.flow_from_directory(r"/content/drive/MyDrive/dataset/training_dataset",target_size = (128,128),
                                             class_mode = "binary",batch_size = 32)
```

```
Found 442 images belonging to 2 classes.
```

```
[5] #Applying ImageDataGenerator functionality to testset
    x_test = test_data.flow_from_directory(r"/content/drive/MyDrive/dataset/test_dataset",target_size = (128,128),
                                             class_mode = "binary",batch_size = 32)
```

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

7.2 Feature 2

Model Building:

After the preprocessing of training dataset using keras ImageDataGenarator,the preprocessed data used to build the deep learning model.For forest fire detection,the Convolutional Neural Network(CNN) algorithm is used.Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification.The Convolutional Neural Network has several layers such as Convolutional layer, Pooling layer, Flatten layer, Dense layer.After defining all the layers ,the model is compiled and built with 10 number of epochs.

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

```
[ ] #Training the model
    model.fit(x_train, epochs = 10, validation_data=x_test, steps_per_epoch=len(x_train), validation_steps=len(x_test))
```

```
Epoch 1/10
14/14 [==============================] - 116s 8s/step - loss: 3.3361 - accuracy: 0.5814 - val_loss: 0.5760 - val_accuracy: 0.8347
Epoch 2/10
14/14 [==============================] - 28s 2s/step - loss: 0.4488 - accuracy: 0.8145 - val_loss: 0.1567 - val_accuracy: 0.9256
Epoch 3/10
14/14 [==============================] - 28s 2s/step - loss: 0.2636 - accuracy: 0.8869 - val_loss: 0.1218 - val_accuracy: 0.9421
Epoch 4/10
14/14 [==============================] - 31s 2s/step - loss: 0.2231 - accuracy: 0.8959 - val_loss: 0.1011 - val_accuracy: 0.9587
Epoch 5/10
14/14 [==============================] - 28s 2s/step - loss: 0.1899 - accuracy: 0.9140 - val_loss: 0.1215 - val_accuracy: 0.9669
Epoch 6/10
14/14 [==============================] - 30s 2s/step - loss: 0.1745 - accuracy: 0.9231 - val_loss: 0.1237 - val_accuracy: 0.9669
Epoch 7/10
14/14 [==============================] - 33s 2s/step - loss: 0.1999 - accuracy: 0.9072 - val_loss: 0.1295 - val_accuracy: 0.9587
Epoch 8/10
14/14 [==============================] - 33s 2s/step - loss: 0.1967 - accuracy: 0.9163 - val_loss: 0.0811 - val_accuracy: 0.9835
Epoch 9/10
14/14 [==============================] - 28s 2s/step - loss: 0.1897 - accuracy: 0.9140 - val_loss: 0.1058 - val_accuracy: 0.9421
Epoch 10/10
14/14 [==============================] - 28s 2s/step - loss: 0.1859 - accuracy: 0.9253 - val_loss: 0.0943 - val_accuracy: 0.9752
<keras.callbacks.History at 0x7f6731527b10>
```

```
[ ] #Save the model
    model.save("forestfire.h5")
```

7.3 Feature 3

Alert System:

In this forest fire detection model,an alert system is developed so that if the fire is detected by the model , then automatically an SMS alert is sent to the user regarding the forest fire.The SMS alert system can be achieved by Twilio Cloud Communications.Twilio is a customer engagement platform used by hundreds of thousands of businesses and more than ten million developers worldwide to build unique, personalized experiences for their customers

Twilio Service

```
[ ] account_sid = 'ACff1bf8e870b4b3af4ccdfe1e91b3842e'
    auth_token = '3eb9c1342edc25b81ab9b736b45f754d'
    client = Client(account_sid, auth_token)

    message = client.messages \
        .create(
             body='Alert: Forest fire is detected!!',
             from_=' +14793973433',
             to='+918940861869'
         )

    print(message.sid)
```

```
SMa3cf62847c585d596a7af8f9c26c5f53
```

7.4 Feature 4

Video Analysis:

As in this CNN model,the streaming videos are imported as input.The videos should convert into frames and those frames are reshaped corresponding to the model.This can be achieved by OpenCV.OpenCV is the huge open-source library for the computer vision, machine learning, and image processing .By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

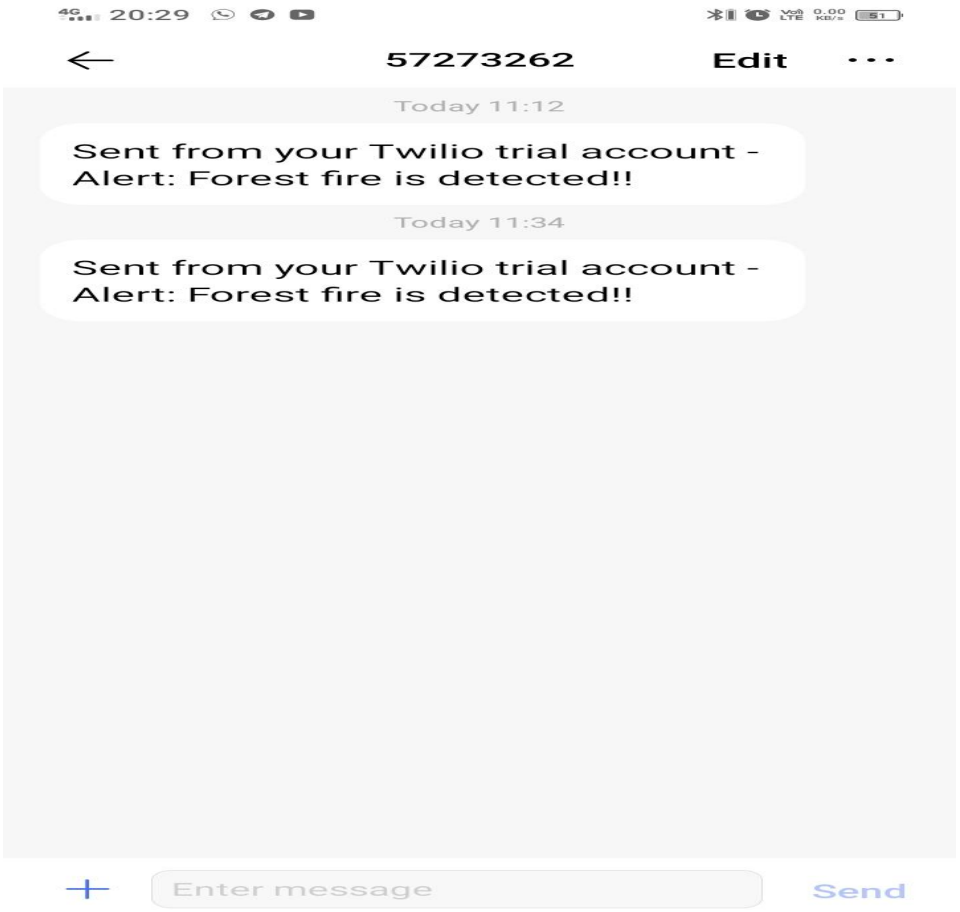
In [ ]:
```
model=load_model(r'forestfire.h5')
video=cv2.VideoCapture("sample_video.mp4")
while(1):
    success, frame=video.read()
    cv2.imwrite("image.jpg", frame)
    img=keras.utils.load_img("image.jpg", target_size=(128,128))
    x=keras.utils.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    pred = model.predict(x)
    cv2.imshow("frame",frame)
    if cv2.waitKey(10000) & 0xFF== ord('a'):
        break
    if pred[0]==1:
        account_sid = 'ACff1bf8e870b4b3af4ccdfe1e91b3842e'
        auth_token = '3eb9c1342edc25b81ab9b736b45f754d'
        client = Client(account_sid, auth_token)

        message = client.messages \
            .create(
                 body='Alert: Forest fire is detected!!',
                 from_=' +14793973433',
                 to='+918940861869'
        )

        print(message.sid)
        print('Fire Detected')
        print ('SMS sent!')
        break
    else:
        print("No Danger")

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

CHAPTER - 8
TESTING

8.1 TEST CASES.

- To check whether the model can predict the forest fire based on the input video given.
- If the fire is detected the model will play alarm sound, send an alert message through twilio account stating “Alert: Forest fire is detected” to the registered mobile numbers.
- If the fire is not detected it will not play the alarm sound, it will not send any alert messages to the registered mobile numbers.

8.2 User Acceptance Testing



CHAPTER - 9
RESULTS

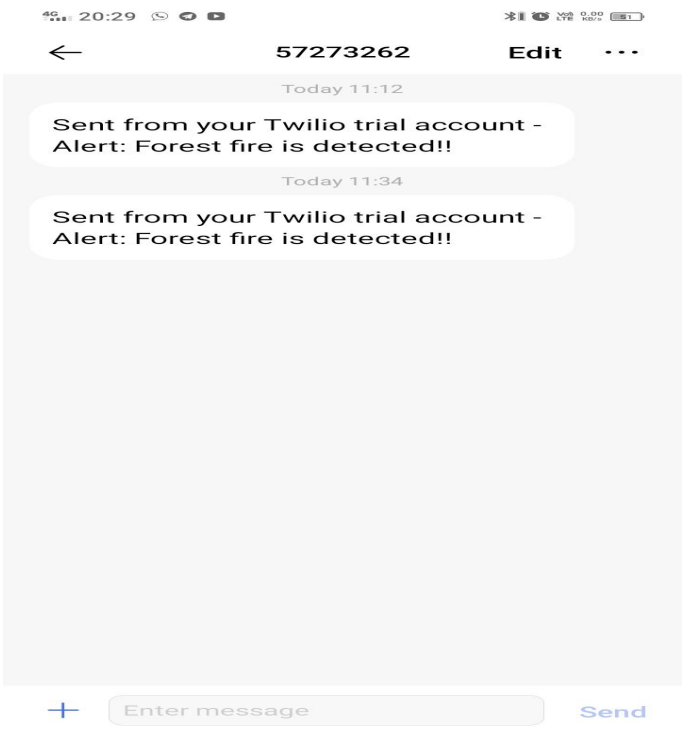
9.1 PERFORMANCE METRICS.

Using this any fire in the forest can be detected accurately and also an efficient alert is sent to the forest fire department (officers) so that they can reach the exact spot on time and the fire can be stopped/ prevented from further spreading.

Sample images of the model output:

If the fire is detected:

```
1/1 [==============================] - 0s 72ms/step
SM36a9b67288044142ddf5938ca52a8cd2
Fire Detected
Alarm sound is playing
SMS sent!
```



CHAPTER - 10

ADVANTAGES & DISADVANTAGES

Advantages:

- Reports to the forest department and nearest police station quickly so as to take faster actions
- Automated analysis of fire detection.
- Improved accuracy rate.
- Reduced time for computational complexity.

Disadvantages:

- Cost for fixing the cameras all over the forest is expensive.
- System requirement for running the model is high.
- Lots of training data is required.
- Overfitting is a major challenge while training the model.

CHAPTER - 11

CONCLUSION

By using this model which is based on Convolution Neural Network (CNN) one can accurately detect whether there is a fire or not in the forest based on the image sets that were used to train this model. Moreover accuracy achieved using this model is very high when compared to previous techniques that were used in the process of forest fire detection.

CHAPTER - 12

FUTURE SCOPE

- An user interface application can be built by integrating this CNN model into it in the future, which can be accessed by all the forest fire departments in an efficient way.
- An advancement of live video broadcasting in case of forest fire being detected through the application can also be achieved.
- By fixing GPS trackers with the cameras the location of the fire area will be located exactly.

CHAPTER - 13
APPENDIX

Source Code:

```
#Importing the ImageDataGenerator class
import keras
from keras.preprocessing.image import ImageDataGenerator

#import model building libraries
#To import Sequential
from keras.models import Sequential
#To add Dense layers
from keras.layers import Dense
#To add Convolution layer
from keras.layers import Convolution2D
#To add  Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')

#importing modules for video analysis
import keras
import tensorflow as tf
import cv2
import numpy as np
from subprocess import Popen
from keras.preprocessing import image
from keras.models import load_model
from twilio.rest import Client
from playsound import playsound

#Define the parameters/arguments for ImageDataGenerator class
train_data = ImageDataGenerator(rescale= 1./255,shear_range=
   0.2,rotation_range= 180,horizontal_flip = True,vertical_flip =
   True,zoom_range = 0.2)
test_data = ImageDataGenerator(rescale= 1./255)
#Applying ImageDataGenerator functionality to trainset
x_train =  train_data.flow_from_directory(r
   "/content/drive/MyDrive/dataset/training_dataset",target_size =
```

```
   (128,128),
              class_mode = "binary",batch_size = 32)
#Applying ImageDataGenerator functionality to testset
x_test =
   test_data.flow_from_directory(r"/content/drive/MyDrive/dataset/te
   st_dataset",target_size = (128,128),
    class_mode = "binary",batch_size = 32)

#Initializing the model
model=Sequential()
#add convolutional layer
model.add(Convolution2D(32,(3,3),activation = "relu",input_shape =
   (128,128,3)))
#add maxpooling layer
model.add(MaxPooling2D(pool_size = (2,2)))
#add flatten layer
model.add(Flatten())
#add hidden layer
model.add(Dense(300,activation='relu'))
model.add(Dense(300,activation='relu'))
#add output layer

model.add(Dense(1, activation="sigmoid"))
#configure the learning process
model.compile(loss="binary_crossentropy", metrics=["accuracy"],
   optimizer='adam')
#Training the model
model.fit(x_train, epochs = 10, validation_data=x_test,
   steps_per_epoch=len(x_train), validation_steps=len(x_test))
#Save the model
model.save("forestfire.h5")


#detection of fire from video
model=load_model(r'forestfire.h5')
video=cv2.VideoCapture("sample_video.mp4")
while(1):
    success, frame=video.read()
    cv2.imwrite("image.jpg", frame)
    img=keras.utils.load_img("image.jpg", target_size=(128,128))
    x=keras.utils.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    pred = model.predict(x)
```

```
    cv2.imshow("frame",frame)
    if cv2.waitKey(10000) & 0xFF== ord('q'):
        break
    if pred[0]==1:
        account_sid = 'ACbc91e3dd3e67b221d468f5b76b3151d3'
        auth_token = '401857e61dbe714cc6e7b25aa63fb937'
        client = Client(account_sid, auth_token)

        message = client.messages \
           .create(
                body='Alert: Forest fire is detected!!',
                from_='+16295295924',
                to='+919360233080'
            )

        print(message.sid)
        print('Fire Detected')
        print('Alarm sound is playing')
        playsound(r'Siren.mp3')
        print ('SMS sent!')
        break
    else:
        print("No Danger")

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

GitHub & Project demo link:

Demonstration Video Link : https://youtu.be/nfRlpi938ow

GitHub project link:

Emerging Methods For Early Detection Of Forest Fires

Link: https://github.com/IBM-EPBL/IBM-Project-12494-1659452250