**Professional Readiness for Innovation, Employability and Entrepreneurship**

Executed By: IBM

# Emerging Methods For Early Detection of Forest Fire

College Name: Anna University Regional Campus-Madurai

Batch: B1-M3E

Team ID: PNT2022TMID47488

Team Leader: Saravana Babu S

Team Members:

Eswari E

Karishma P

Thahir Ahamed T

Faculty Mentor Name: Dr.B. Anandhi Meena

Industry Mentor Name: Shanthi

# CONTENTS

# 1. Introduction

## 1.a) Project Overview

This project helps the giant corporation or a government to monitor the huge reserve forest. The forest fire is detected by using computer vision algorithm baes cameras. These cameras continously monitor the forest and the details were given to the IBM Cloud Server. If there is fire thr respective departments (Fire Department, Forest Department) will get a notification from the server. By using this we can detect and prevent forest fire.

## 1.b) Purpose

The purpose of this project is to provide a timely information about the forest fire. It reduces the number of areas affected by this fire and thereby minimizes the costs of extinguishing the damage caused in the woods and save wildlives. It also monitors the potential damage as well as the cost of firefighting.

## 2. Literature Survey

### 2.a) Existing Problem

Forest fires have become a frequent and most dangerous natural disaster as a result of climate change. The fact that more than 20% of complete world $CO_2$ emissions comes from forest fires indicates that it is a phenomenon which has to be dealt with great attention.

### 2.b) References


[1] Image Processing Based Forest Fire Detection. Vipin V

[2] Sousa, Joaquim & Gamboa, Pedro. (2020). Aerial Forest Fire Detection and Monitoring Using a Small UAV. KnE Engineering. 242-256. 10.18502/keg.v5i6.7038.

[3] Li Jie, & Xiao Jiang. (2009). Forest fire detection based on video multi-feature fusion. 2009 2nd IEEE International Conference on Computer Science and Information Technology. doi:10.1109/iccsit.2009.5234862

[4] X. Zhang and S. Xu, "Research on Image Processing Technology of Computer Vision Algorithm," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), 2020, pp. 122-124, doi:10.1109/CVIDL51233.2020.00030.

[5] S. Kumar et al., "Human-Inspired Camera: A Novel Camera System for Computer Vision," 2021 18th International SoC Design Conference

(ISOCC), 2021, pp. 29-30, doi: 10.1109/ISOCC53507.2021.9613914.
[6]A scientometric analysis and critical review of computer vision applications for construction. Pablo Martinez1 Mohamed AlHussein, Rafiq Ahmad.
[7] An Introduction to Convolutional Neural Networks. Keiron O'Shea1 and Ryan Nash2
[8]Albawi S, Bayat O, Al-Azawi S, Ucan ON. Social Touch Gesture Recognition Using Convolutional Neural Network. Comput Intell Neurosci. 2018 Oct 8;2018:6973103

## 2c.) Problem Statement Definition

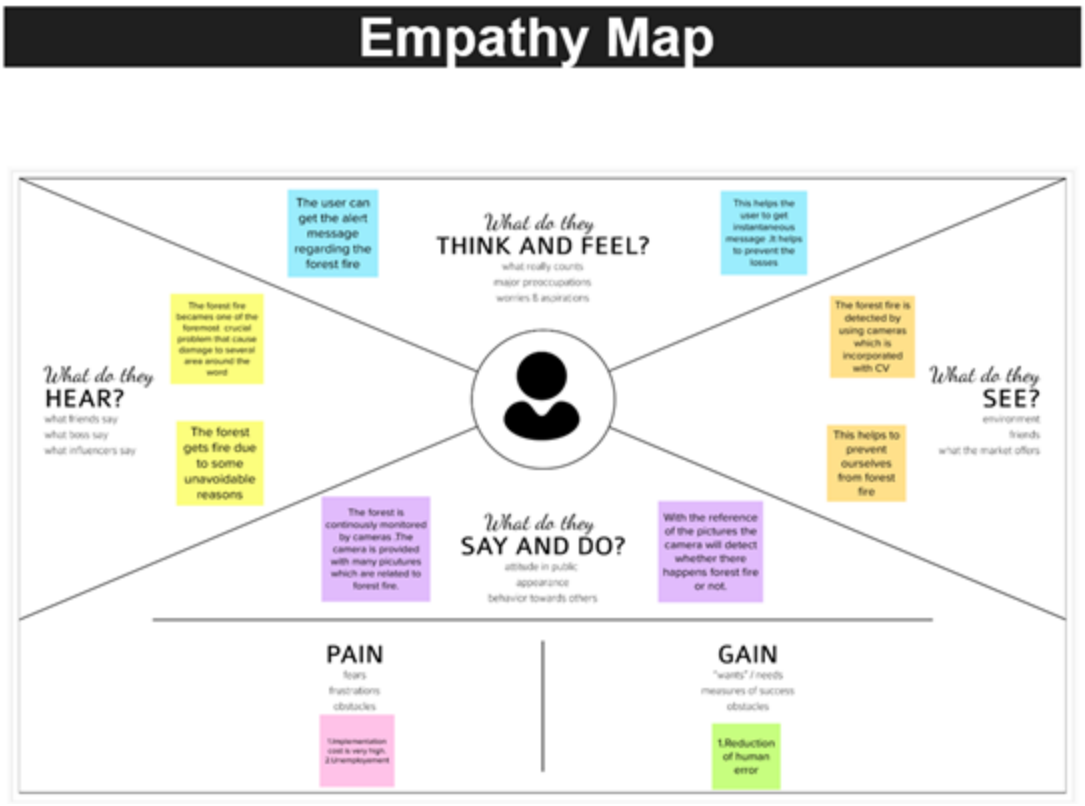
Forest fire are a major environmental issues creating economic and ecological damage while endangering human lives. There are typical about 100,000 wild fires every year over 9 million acres of land have been destroy due to treacherous wild fires.

**Customer Problem Statement:**

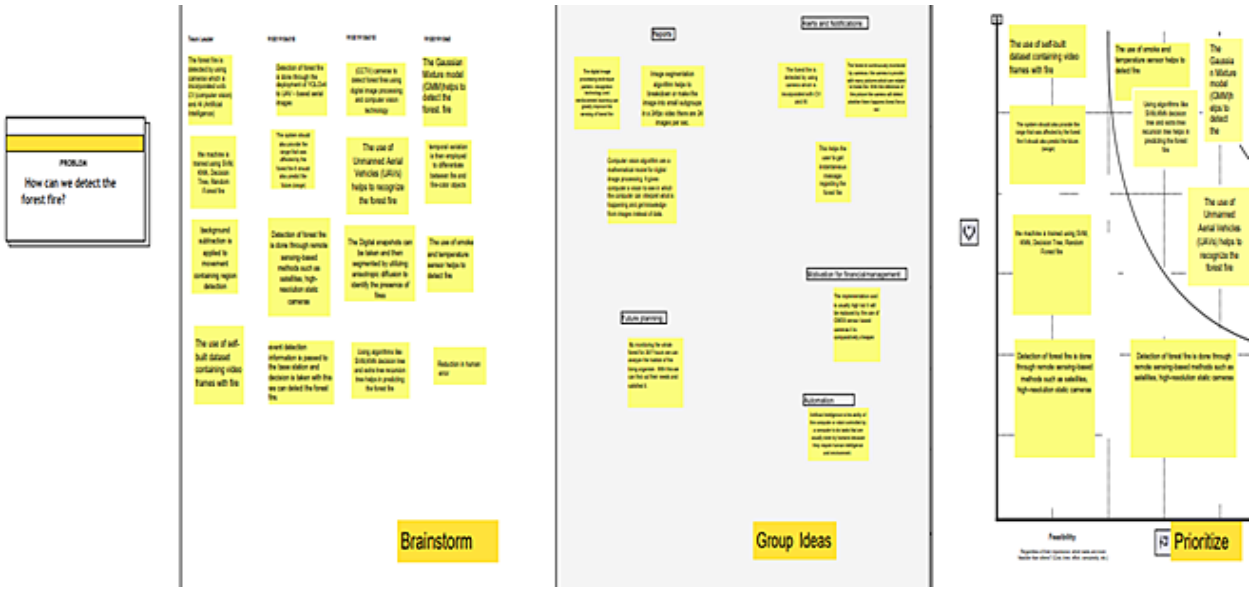
| Problem Statement (PS) | I am (Customer) | I'm trying to | But | Because | Which makes me feel |
|---|---|---|---|---|---|
| PS-1 | Forest Ranger | Detect forest fire | The surveillance of forest is difficult | The sheer volume of forest makes difficult to supervise | Frustrated |
| PS-2 | Tribal People | To survive | The survival in the forest has becoming challenging because of forest fires. | The sudden climate change makes the weather unpredictable and disasters like bush fire and forest fire becomes common | Scared and anxious |
| PS-3 | NGOs based on animal welfare | To save wild lives and trying to give a safer environment | Rescue of endangered species in the forest fires is difficult | The intensity of fire and smoke in the forest area makes difficult to get inside the forest to rescue them | helplessness |

# 3. **Ideation and Proposed Solution**

## 3.a) Empathy Map Canvas



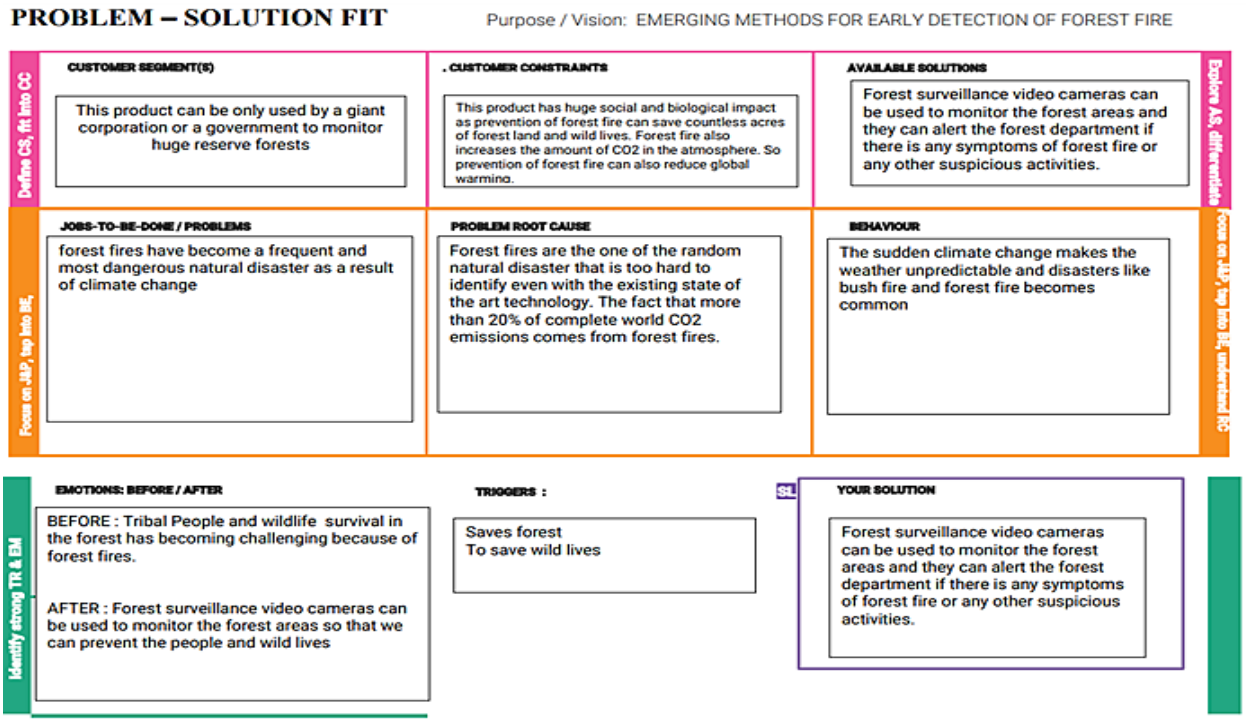

## 3.b) Ideation and Brainstorming



## 3.c) Proposed Solution

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | Emerging methods for early detection of forests fires |
| 2. | Idea / Solution description | Forest surveillance video cameras can be used to monitor the forest areas and they can alert the forest department if there is any symptoms of forest fire or any other suspicious activities. |
| 3. | Novelty / Uniqueness | The digital image processing technique, pattern-recognition technology and reinforcement learning can greatly improve the sensing of forest fire and they are much more effective as they improve forecast and reaction time is much less |
| 4. | Social Impact / Customer Satisfaction | This product has huge social and biological impact as prevention of forest fire can save countless acres of forest land and wild lives. Forest fire also increases the amount of CO2 in the atmosphere. So prevention of forest fire can also reduce global warming. |
| 5. | Business Model (Revenue Model) | This product can be only used by a giant corporation or a government to monitor huge reserve forests. This can be considered as a profitable and useful product as government spends millions of dollars for detection of forest fires and millions more if there is any actual forest fire like rescuing and stopping the fire. |
| 6. | Scalability of the Solution | It is highly challenging to implement this method as we have to install hundreds of cameras to cover a respectable amount of area. The cameras need to be connected to electricity and they also need to be connected to internet to process the image and analyze them. They need to be connected to local server which should be located at the middle of the forest. |

## 3.d) Problem Solution Fit




# 4. **Requirement Analysis**

## 4.a) Functional Requirements

**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 the registered government ID |
| FR-2 | User Confirmation | Confirmation via OTP |
| FR-3 | Overall Surveillance Report | Helps to understand the current scenario in the forest by giving report as “no fire” or “negative”. |
| FR-4 | Cloud Server Access | To save and run the model from the camera footage |
| FR-5 | Live Camera Feed | Real-time monitoring by the forest officials |
| FR-6 | GSM Module | To alert the nearest forest range officer and the local fire department |

# 4.b) Non-Functional Requirements

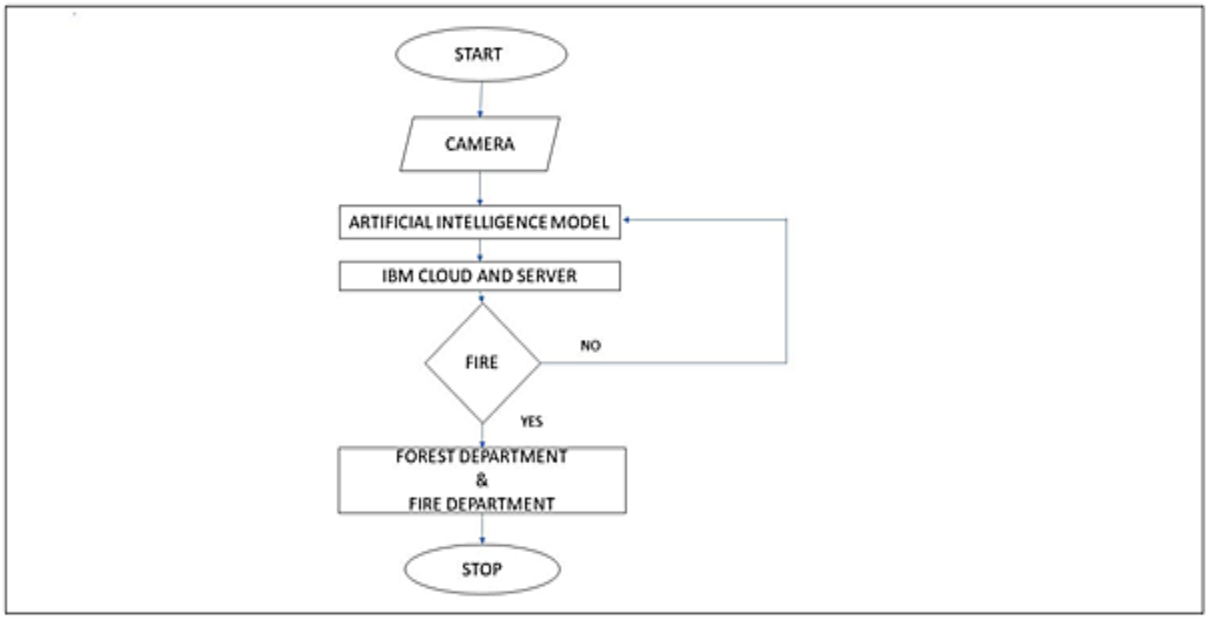
**Non-functional Requirements:**

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | This project as a service can be utilized by the governments who control reserve forests, giant corporations who controls acres of land in which they cultivate trees for commercial purposes and NGOs who try to protect forests and it can also be used by NGOs and forest department to monitor behavior of endangered animals. |
| NFR-2 | **Security** | In order to ensure security in the monitoring process the server is used as IBM cloud which has very good encryption standard. These files can be only accessed by officials from the government of corporations. There will be further security check by OTP for confirmation. The backup videos will be saved in the IBM cloud server. |
| NFR-3 | **Reliability** | The project is very much reliable compared to an previous generation open-source forest monitoring system where the data |

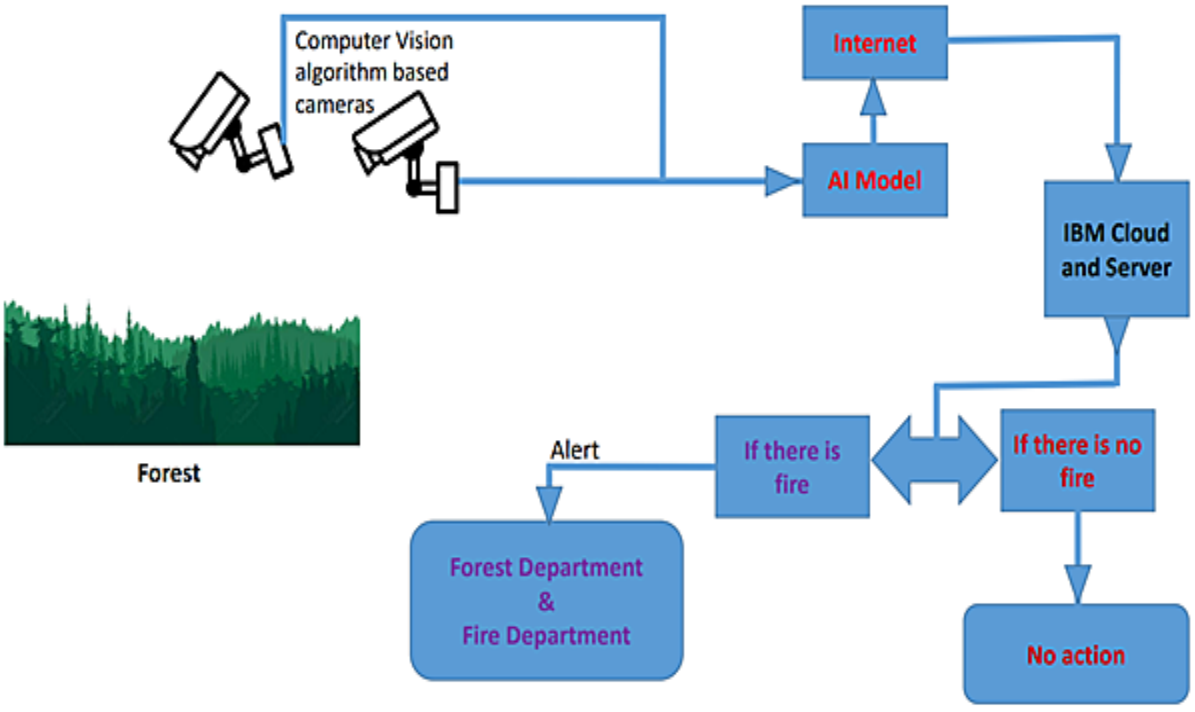
| | | |
|---|---|---|
| | | can be easily manipulated and this is much robust as the initial cost is higher while there will be no need for any maintenance cost |
| NFR-4 | **Performance** | This project works better compared to other methods to detect forest fires like satellite monitoring, IOT sensors or use of IR sensor-based cameras. The accuracy of this model also increases over the period of time. |
| NFR-5 | **Availability** | This data can be only accessed by the officials as this is very sensitive information regarding thousands of acres of forest lands. So this can be opened anywhere by the authorized person as the AI model is connected in IBM server. |
| NFR-6 | **Scalability** | The initial cost to setup is expensive compared to other methods while there will be less or no maintenance cost and the cost to stop a forest fire and the pollution and wildlife lost is much significant compared to the initial setup costs. The project can be easily scaled to larger parts of the forests as they are much simpler to implement. |

## 5. **Project Design**

### 5.a) Data Flow Diagram




### 5.b) Solution and Technical Architecture



## 5.c) User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Forest fire department | Registration | USN-1 | This product can be only used by a giant corporation or a government to monitor huge reserve forest. | It can accept only by government | High | Sprint-1 |
| | Login | USN-2 | Forest fire department staffs are provided with an application with unique login ID and password .By using that they can monitor the forest. | I can access my account | High | Sprint-3 |
| | IBM cloud server | USN-3 | The forest fire is detected using computer vision algorithm based cameras.This cameras continuously monitor the forest and details were given to the IBM cloud server. | I can receive the information | High | Sprint-2 |
| | | USN-4 | The forest fire department staff fetch the details through IBM cloud server | I can receive the information | High | Sprint-4 |
| | | USN-5 | If there is no fire, the department did not get the notification from server. | | High | Sprint-3 |
| | | USN-6 | If there is fire ,the department gets notification from server and corresponding action will be taken. | I can take action | | Sprint-3 |

# 6. **Project Planning and Scheduling**

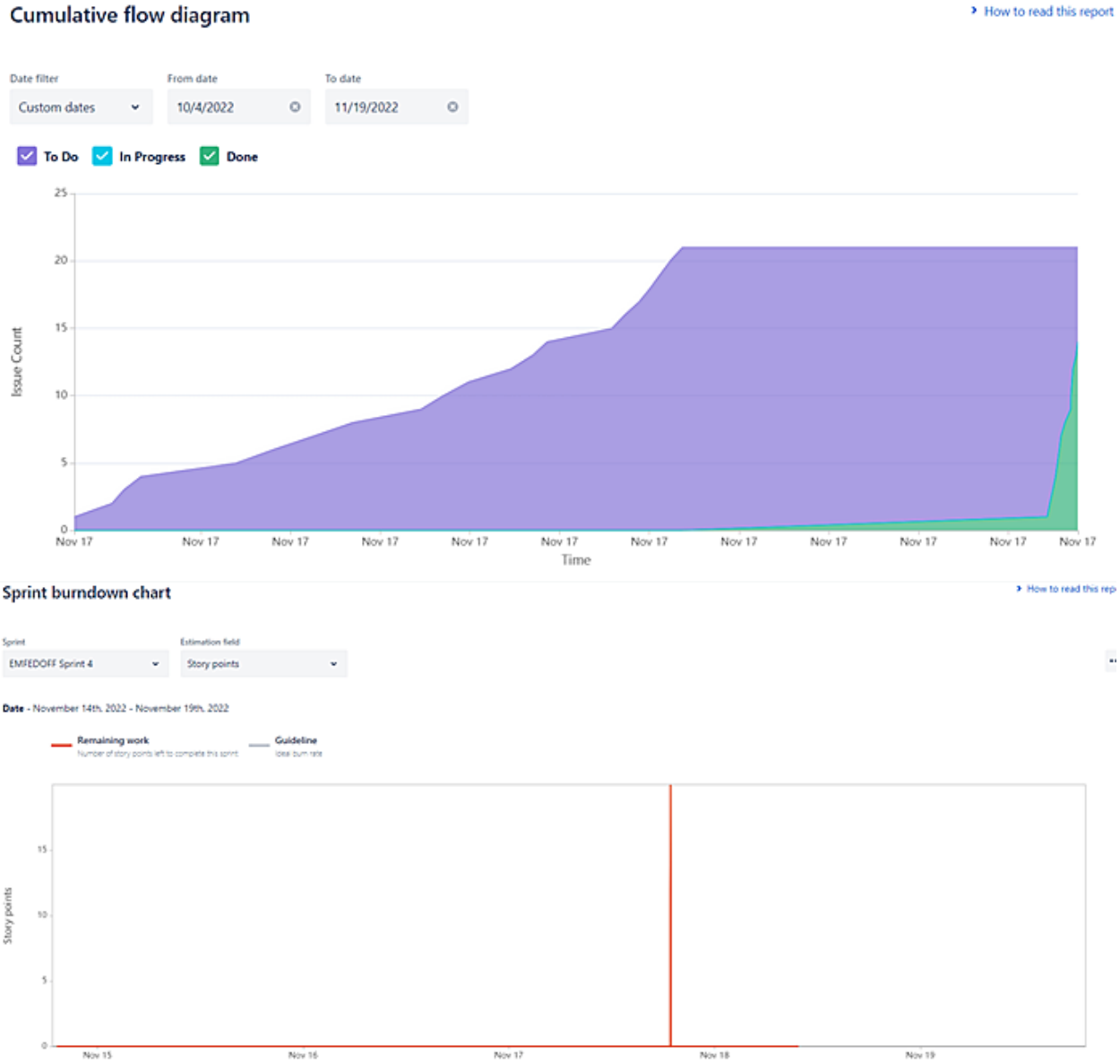
## 6.a) Sprint Planning and Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User story/ Task | Story Points | Priority | Team members |
|---|---|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | As a user, I should be able to register myself in the application by giving my email ID, phone number, username and password | 2 | High | Saravana Babu S |
| Sprint-1 | Login | USN-2 | As a user, I should be able to access my account using my username and password | 2 | High | Saravana Babu S |
| Sprint-1 | Data Collection | USN-3 | Collection of data (images) for testing and training data | 1 | High | Eswari E |
| Sprint-1 | Data preprocessing | USN-4 | Image preprocessing and image segregation for the ease of training AI model and helps to classify images. | 2 | Medium | Karishma P |
| Sprint-2 | Model Training | USN-1 | Training an AI model using the preprocessed data using AI techniques like CNN, RNN, and YOLO Algorithms. | 1 | Medium | Thahir Ahamed T |
| Sprint-2 | | USN-2 | Computer Vision (OpenCV) for video processing and techniques like frame-to-frame segmentation, image interpolation. | 2 | Medium | Thahir Ahamed T |
| Sprint-3 | Local Implementation | USN-1 | Integration of the AI model and the webpage for user interface | 2 | Medium | Saravana Babu S |
| Sprint-3 | | USN-2 | Activation of Twilio account and integration of Twilio account with Open CV and the previously integrated AI model | 1 | High | Eswari E |
| Sprint-3 | | USN-3 | Testing of SMS alert by implementing the entire model locally using Python-Flask | 1 | Medium | Karishma P |
| Sprint-4 | Cloud Deployment | USN-1 | Creation of the IBM Cloud services which are being used in this project. | 2 | High | Saravana Babu S |
| Sprint-4 | | USN-2 | Configuration of the IBM Cloud depending on the project | 1 | Medium | Saravana Babu S |
| Sprint-4 | | USN-3 | Training of the AI model in the cloud with the preprocessed data and saving them. | 1 | High | Eswari E |
| Sprint-4 | | USN-4 | Integrating the cloud trained AI model with webpage, OpenCV and Twilio API services. | 2 | High | Karishma P |
| Sprint-4 | | USN-5 | Final deployment of the entire integrated model | 2 | High | Saravana Babu S, Eswari E, Karishma P, Thahir Ahamed T |

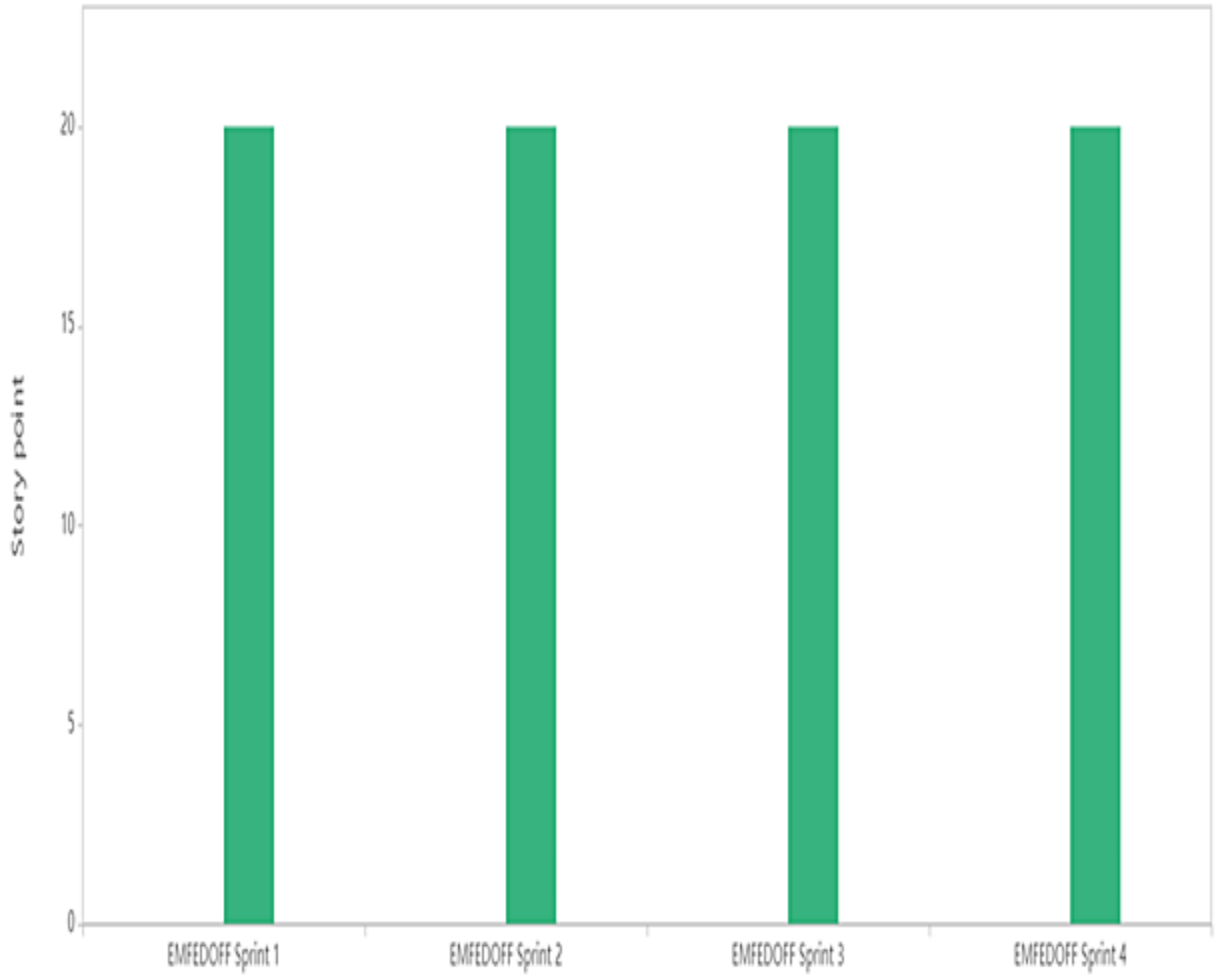
## 6.b) Sprint Delivery Schedule

| 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 |

## 6.c) Reports from JIRA



The amount of work in the sprint when it began. The amount of work done during the sprint.




| Sprint | Commitment | Completed |
|---|---|---|
| EMFEDOFF Sprint 1 | 0 | 20 |
| EMFEDOFF Sprint 2 | 0 | 20 |
| EMFEDOFF Sprint 3 | 0 | 20 |
| EMFEDOFF Sprint 4 | 0 | 20 |

# 7. **Coding and Solution**

## 7.a) Feature 1 (Model Building)

### Coding

```
pwd
```

```
'/home/wsuser/work'
```

```
!pip install keras
!pip install tensorflow
```

```
Output exceeds the size limit. Open the full output data in a text editor
Requirement already satisfied: keras in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.0)
Requirement already satisfied: tensorflow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.2)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.0)
Requirement already satisfied: protobuf>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: tensorboard~=2.7 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: absl-py>=0.4.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.12.0)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.20.3)
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: six>=1.12.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.15.0)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: wheel<1.0,>=0.32.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.37.0)
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (4.1.1)
```

```
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (4.1.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.23.1)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.2.1)
Requirement already satisfied: keras<2.8,>=2.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (0.4.4)
Requirement already satisfied: werkzeug>=0.11.15 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (2.0.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.7->tensorflow) (1.23.0)
...
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (2022.9.24)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (2.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard~=2.7->tensorflow) (3.2.1)
```

```
!pip install numpy
!pip install matplotlib
!pip install pandas
```

```
Requirement already satisfied: numpy in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.20.3)
Requirement already satisfied: matplotlib in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (3.5.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: packaging>=20.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (21.3)
Requirement already satisfied: numpy>=1.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (1.20.3)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (3.0.4)
```

```
Requirement already satisfied: numpy>=1.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (1.20.3)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: cycler>=0.10 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil>=2.7->matplotlib) (1.15.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.3.4)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
```

```
    # import keras library
    import keras
    #import ImageDataGenerator from keras.preprocessing.image
    from keras.preprocessing.image import ImageDataGenerator
```

```
    from tensorflow import keras
    from tensorflow.keras.preprocessing import image_dataset_from_directory
```

```
    #Import the Model Building Libraries
    import warnings
    warnings.filterwarnings('ignore')
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
    from tensorflow.keras.layers import Convolution2D
```

```
import warnings
warnings.filterwarnings('ignore')
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

```
#Preprocessing the training image dataset _data_augmentation
train_datagen = ImageDataGenerator(
                    rotation_range=180,
                    brightness_range=None,
                    shear_range=0.4,
                    zoom_range=0.3,
                    horizontal_flip=True,
                    vertical_flip=True,
                    rescale=1./255,)
```

```
# Preprocessing the Testing Dataset Images _data_augmentation
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
import types
from botocore.client import Config
import ibm_boto3
```

```
def __iter__ (self): return 0
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='c54TjAL2Fk4lME_t5N7QcuCdFS5ZEsRYBTbvEJIKibqu',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'forestfireimageclassification-donotdelete-pr-yuifovm4ejm0le'
object_key = 'Dataset.zip'

streaming_body_10 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
!pip install -U ibm-watson-machine-learning
```

Python

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.3.4)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning)
(2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-
machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*-
>ibm-watson-machine-learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm-watson-machine-learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-learning) (3.0.4)
```

```
import io
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_10.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

Python

```
pwd
```

Python

```
'/home/wsuser/work'
```

```
import os
filenames= os.listdir ('/home/wsuser/work/Dataset')
```

Python

```
xtrain = train_datagen.flow_from_directory('/home/wsuser/work/Dataset/train_set',
                                           target_size=(64,64),
                                           class_mode='binary',
                                           batch_size=100)
```

Python

```
Found 436 images belonging to 2 classes.
```

```
xtest = train_datagen.flow_from_directory('/home/wsuser/work/Dataset/test_set',
                                          target_size=(64,64),
```

```
xtest = train_datagen.flow_from_directory('/home/wsuser/work/Dataset/test_set',
                                          target_size=(64,64),
                                          class_mode='binary',
                                          batch_size=100)
```

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

```
xtrain.class_indices
```

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

```
#Initialize the Model
model = Sequential()
```

```
#Convolution Layer
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

```
#MaxPooling Layer
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
#Flatten Layer
model.add(Flatten())
```

```
#Adding the Dense Layer
#Hidden Layer
model.add(Dense(350,activation='relu')) # Hidden layer 1
model.add(Dense(200,activation='relu')) # Hidden layer 2
model.add(Dense(2, activation='relu')) # Hidden layer 3
```

```
#Output Layer
model.add(Dense(1,activation='softmax'))
```

```
#Configuring the Learning Prcoess
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

```
#Training the Model
model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=10,
                    validation_data=xtest,
                    validation_steps=len(xtest))
```

```
Epoch 1/10
5/5 [==============================] - 20s 4s/step - loss: 0.6933 - accuracy: 0.3555 - val_loss: 0.6927 - val_accuracy: 0.4050
Epoch 2/10
5/5 [==============================] - 16s 3s/step - loss: 0.6922 - accuracy: 0.3555 - val_loss: 0.6923 - val_accuracy: 0.4050
Epoch 3/10
5/5 [==============================] - 16s 3s/step - loss: 0.6915 - accuracy: 0.3555 - val_loss: 0.6918 - val_accuracy: 0.4050
Epoch 4/10
5/5 [==============================] - 16s 3s/step - loss: 0.6909 - accuracy: 0.3555 - val_loss: 0.6914 - val_accuracy: 0.4050
Epoch 5/10
5/5 [==============================] - 16s 3s/step - loss: 0.6902 - accuracy: 0.3555 - val_loss: 0.6909 - val_accuracy: 0.4050
Epoch 6/10
5/5 [==============================] - 16s 4s/step - loss: 0.6895 - accuracy: 0.3555 - val_loss: 0.6905 - val_accuracy: 0.4050
Epoch 7/10
5/5 [==============================] - 16s 3s/step - loss: 0.6889 - accuracy: 0.3555 - val_loss: 0.6901 - val_accuracy: 0.4050
Epoch 8/10
5/5 [==============================] - 15s 3s/step - loss: 0.6882 - accuracy: 0.3555 - val_loss: 0.6898 - val_accuracy: 0.4050
Epoch 9/10
5/5 [==============================] - 15s 3s/step - loss: 0.6877 - accuracy: 0.3555 - val_loss: 0.6894 - val_accuracy: 0.4050
Epoch 10/10
5/5 [==============================] - 16s 3s/step - loss: 0.6871 - accuracy: 0.3555 - val_loss: 0.6890 - val_accuracy: 0.4050

<keras.callbacks.History at 0x7fb6741addf0>
```

```
#Save the Model
model.save('Forest_fire.h5')
```

```
!tar -zcvf image-classification-model_new.tgz Forest_fire.h5
```
[27]

```
Forest_fire.h5
```

```
ls -1
```
[28]

```
Dataset/
Forest_fire.h5
image-classification-model_new.tgz
my_model2.tar.gz
```

```
!pip install watson-machine-learning-client --upgrade
```
[29]

```
Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
```

```
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)
```

```python
from ibm_watson_machine_learning import APIClient
```

Python

```python
wml_credentials={
                 "url": "https://us-south.ml.cloud.ibm.com",
                 "apikey": "7BdOMEdkktNJ6spAgONr8NeM9TQG965MavZlAwFJ3Z5S"
                 }
```

Python

```python
client= APIClient(wml_credentials)
```

Python

```python
def guid_from_space_name(client, space_name ):
    space = client.spaces.get_details()
    #print (space)
    return (next(item for item in space ['resources'] if item ['entity']['name']== space_name)['metadata']['id'])
```

```python
space_uid = guid_from_space_name(client, 'Forest_fire_model')
print ('Space ID = ' + space_uid)
```

```
Space ID = 591e1daf-f677-4b5f-9dba-5dba20311381
```

```python
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

```python
client.software_specifications.list()
```

```
Output exceeds the size limit. Open the full output data in a text editor
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
```

```
spark-mllib_3.0-scala_2.12     09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9      0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6          0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                     0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod   1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6              10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl      111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10        125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9             12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6        154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                   1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6         1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6           1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt  1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6           1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod   217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda        26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                       295addb5-9ef9-547e-9bf4-92ae3563e720  base
...
autoai-kb_3.1-py3.7            632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8         634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
-----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

```
[37]
software_spec_uid = client.software_specifications.get_uid_by_name ('tensorflow_rt22.1-py3.9')
software_spec_uid
```

```
[37]
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
model_details= client.repository.store_model (model= 'image-classification-model_new.tgz'
,meta_props={
client.repository.ModelMetaNames.NAME:'CNN',
client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
}
                                                    )
model_id= client.repository.get_model_id (model_details)
```

```
[38]
model_id
```

```
[39]
'3dae6756-d295-4131-b79b-9b22fcc661b8'
```

```
client.repository.download (model_id, 'my_model4.tar.gz')
```

```
[41]
Successfully saved model content to file: 'my_model4.tar.gz'

'/home/wsuser/work/my_model4.tar.gz'
```

# Solution

```
[42]
from keras.models import load_model
from keras.preprocessing import image
```

```
[43]
model= load_model('Forest_fire.h5')
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='c54TjAL2Fk4lME_t5N7QcuCdFS5ZEsRYBTbvEJIKibqu',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'forestfireimageclassification-donotdelete-pr-yuifovm4ejm0le'
```

```
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='c54TjAL2Fk4lME_t5N7QcuCdFS5ZEsRYBTbvEJIKibqu',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'forestfireimageclassification-donotdelete-pr-yuifovm4ejm0le'
object_key = 'forest.jpeg'

streaming_body_12 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
[44]
```

```
!pip install pillow
```

```
[45]
```

```
Requirement already satisfied: pillow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (9.0.1)
```

```
from PIL import Image
img= Image.open(streaming_body_12)
img.show()
```

```
new_image=img.resize((64,64))
```

```
import numpy as np
x= image.img_to_array(new_image)
x= np.expand_dims(x, axis=0)
```

```
pred= model.predict(x)
```

[49]

```
f=int(pred[0])
print(f)
```

[50]

```
1
```

```
index=['Forest without fire','Forest with fire' ]
```

[51]

```
a=index[f]
print(a)
```

[52]

```
Forest with fire
```

# 7.b) Feature 2 (Video Analysis)

## Coding

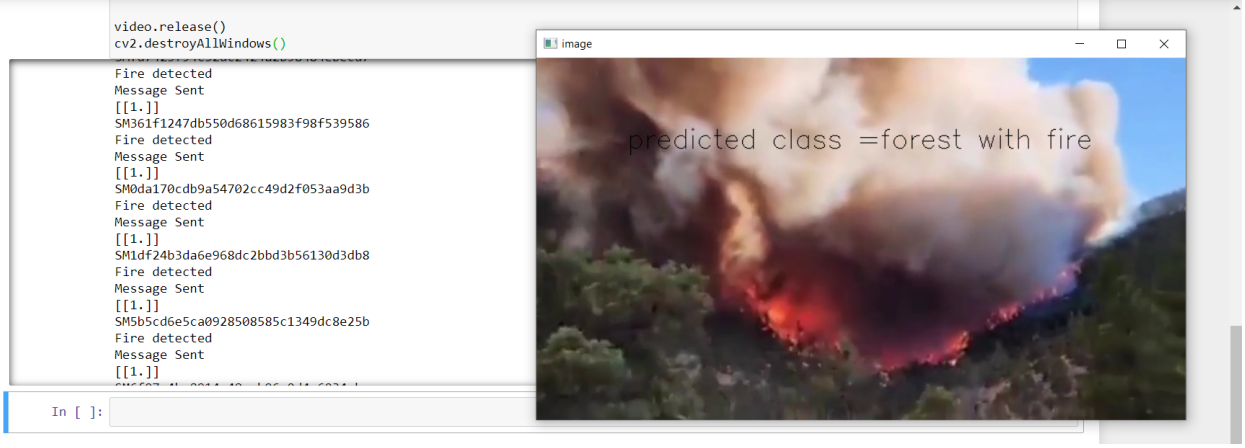
In [1]:
```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import cv2
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from twilio.rest import Client
from playsound import playsound
```

In [2]:
```
model = load_model(r'C:\Users\admin\Downloads\ai\Forest_fire.h5')
video=cv2.VideoCapture(r'C:\Users\admin\Downloads\ai\Forest_fire_1.mp4')
```

In [*]:
```
while (1):
    success, frame= video.read()
    cv2.imwrite('forest.jpg', frame)
    img= image.load_img('forest.jpg', target_size=(64,64,3))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    pred=model.predict(x)
    p= int(pred[0])
    print (pred)
    name=['forest without fire','forest with fire']
    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 = 'AC8d900acfe2f3400cbed2b4a9abbcda90'
      auth_token= '6d169a03f7ef5659901e40f2b5403931'
      client = Client (account_sid, auth_token)
      message = client.messages\
      .create(
      body='Forest Fire is detected, stay alert',
      from_= '+18176702460',
      to='+919962278677')
      print(message.sid)
      print ('Fire detected')
      print ('Message Sent')
      cv2.imshow('image',frame)
    else:
      print ('No Danger')
    cv2.imshow('image',frame)
    if cv2.waitKey(1) & 0xFF == ord('a'):
      break

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

## Solution



# 8. **Testing**

## 8.a) Testing of CNN Model

```
[42]
from keras.models import load_model
from keras.preprocessing import image
```

```
[43]
model= load_model('Forest_fire.h5')
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='c54TjAL2Fk4lME_t5N7QcuCdFS5ZEsRYBTbvEJIKibqu',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'forestfireimageclassification-donotdelete-pr-yuifovm4ejm0le'
```

```
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='c54TjAL2Fk4lME_t5N7QcuCdFS5ZEsRYBTbvEJIKibqu',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'forestfireimageclassification-donotdelete-pr-yuifovm4ejm0le'
object_key = 'forest.jpeg'

streaming_body_12 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
[44]
```

```
!pip install pillow
```

```
[45]
```

```
Requirement already satisfied: pillow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (9.0.1)
```

```
from PIL import Image
img= Image.open(streaming_body_12)
img.show()
```

```
new_image=img.resize((64,64))
```

```
import numpy as np
x= image.img_to_array(new_image)
x= np.expand_dims(x, axis=0)
```

```
[49]
pred= model.predict(x)
```

```
[50]
f=int(pred[0])
print(f)
```

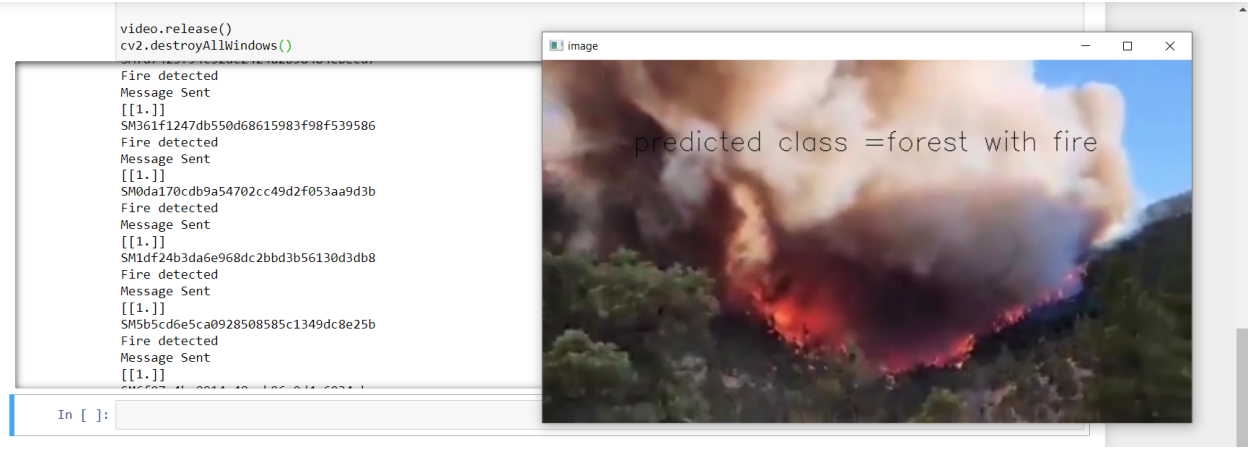
```
...
1
```

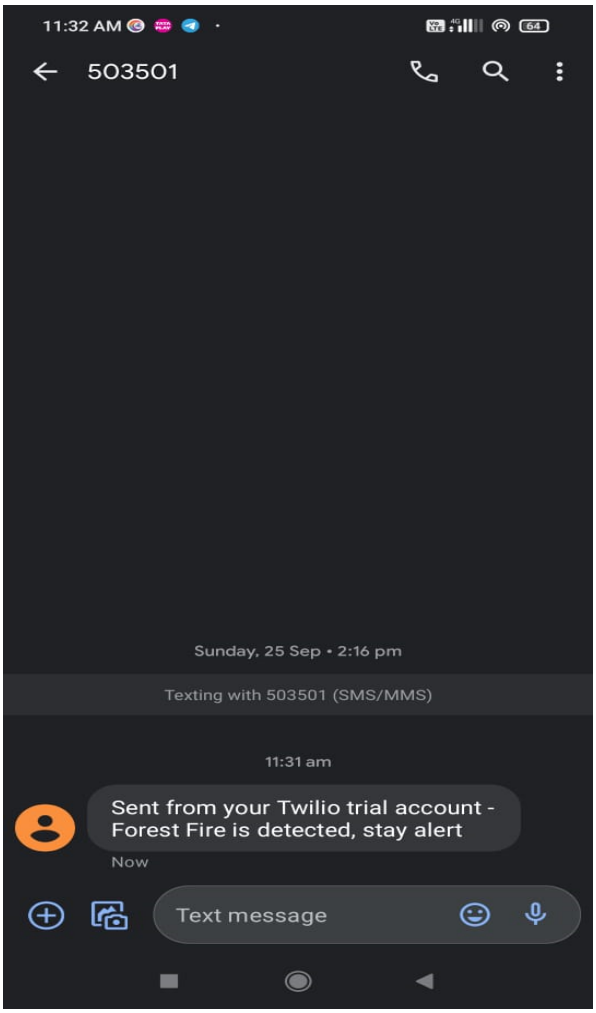
```
[51]
index=['Forest without fire','Forest with fire' ]
```

```
[52]
a=index[f]
print(a)
```

```
...
Forest with fire
```

# 8.a) Testing of Video Analysis with Twilio SMS services



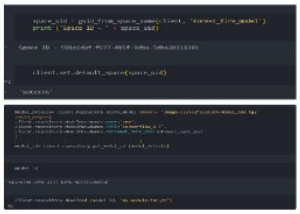
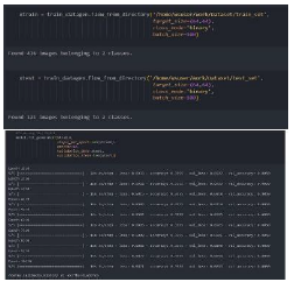



## 8.b) User Acceptance Metrics

| Test case ID | Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Comments | TC for Automation(Y/N) | BUG ID | Executed By |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| model developmen | Functional | Source | the model must be created and must be stored in cloud for local deployment which can be downloaded later using credentials | ibm cloud registration | 1.Enter URL and click go<br>2. Click register for new user<br>3.create account | https://www.ibm.com/in-en/cloud | dashboard will be loaded | Working as expected | Pass | | n | | E.Eswari |
| model developmen | Functional | Process | call the model deployed in cloud with model ID | IBM watson studio must be enabled | enter the code to call the model id and deploy it locally | http://localhost:8888 | when model id is called the id must exist in the space | Working as expected | pass | | | | T.Thahir ahamed |
| LoginPage_TC_OO3 | Functional | registration | code generation and interlinking all the blocks | avaconda google colab jupyter notebook | all codes must be deployed properly step by step and inter contetion between the platforms and twilio must be verified | jupyter code | must run all cases and send sms to recognised | | | | | | P.Karishma |
| LoginPage_TC_OO4 | Functional | registration | authorized person must register their mobile number to get the axcess of alert message | the person must have a vaild mobile number based on his geo location and country | go to twilio | enter number :8220455312 , enter otp : 527945 | number get registered to send alert message | number get registered to send alert message | | | n | | S.Saravana babu |
| LoginPage_TC_OO4 | Functional | Login page | Verify user is able to log into application with InValid credentials | | 2Click on My Account dropdown button<br>3.Enter Valid username/email in Email text box<br>4.Enter Invalid password in password text box 5.Click on the login button | Username: saravanababu413@gmail.com password: Testing1236786867 86876876 | Application should show 'Incorrect email or password ' validation message. | | | | | | |

# 9. **Results**

## 9.a) Performance Metrics

| S. No | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Model Summary | **-model** is developed and stored in IBM Watson studio and it can be called using the model ID and can be deployed locally |  |
| 2. | Accuracy | Training Accuracy – 83%<br><br>Validation Accuracy -75%<br><br>Class Detected - 2 |  |
| 3. | Confidence Score | Confidence Score -79<br><br>Prediction |  |

## 10. **Advantages and Disadvantages**

Adavantage:

- This is helps the giant corporation or a government to monitor the forest
- Forest surveillance video cameras can be used to monitor the forest areas so that we can prevent the people and wild lives
- It helps to save the forest and wild lives
- The digital image processing technique pattern recognition technology and reinforcement learning can greatly improve the sensing of forest fires.

Disadvantage:

- Sometimes we might have errors
- forest fire department have to monitor  the system regularly
- Implementation cost is very high
- loss of valuable wood resources
- Deterioration of catchment areas
- Loss of biodiversity and extermination of flora and fauna.
- Loss of wildlife habitation and exhaustion of wildlife.
- Global warming.

## 11. Conclusion

Forest fires are a very serious problem in many countries, and global warming may contribute to make this problem worse. Experts agree that, in order to prevent these tragedies from happening, it is necessary to invest in new technologies and equipment that enable a multifaceted approach.Since there are many technologies to prevent the forest fire ,I suggest that this technology really helps to control the forest fire.It has good encryption standard.The files can be only assessed by officials.There will be further security check by OTP.So I prefer that this technology really helps to prevent the forest fire and save wildlives.

## 12. Future Scope

This project carries a broad prospective for future. Moreover it is a need for great research to be done inthis field in the coming years. In future, our project can be extended towards finding an efficient way of localization of the fire, gravity of fire, direction of spread, area burnt and many more.Moreover, we can include the region specific meteorological data in the dataset for generating model for prediction.

## 13. **Appendix**

### 13.a) Source Code

https://github.com/IBM-EPBL/IBM-Project-6213-1658824695/tree/main/Final%20Deliverables

### 13.b) Github & Demo Link

Github: https://github.com/IBM-EPBL/IBM-Project-6213-1658824695

Demo Link: https://drive.google.com/file/d/1cunP9mHRUqjFaqoka-avV4Qp_k6NjyA0/view?usp=sharing