EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

USING ARTIFICIAL INTELLIGENCE

A Project report submitted in partial fulfillment of 7th semester in degree Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

Team ID: PNT2022TMID33568

Team Members:

Team Leader	Poorani D	922519106107
Team Member 01	Kaviya S	922519106072
Team Member 02	Prithiba G	922519106116
Team Member 03	Prithika K	922519106117

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING UNIVERSITY COLLEGE OF ENGINEERING,KARUR

ANNA UNIVERSITY: CHENNAI 600025

NOV-2022

V.S.B ENGINEERING COLLEGE, KARUR

(Approved by AICTE & Affiliated by Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report "**EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES**" is the Bonafide record work done by **Ms POORANI D** (**922519106107**), **Ms KAVIYA S** (**922519106072**), **Ms PRITHIBA G** (**922519106116**), and **Ms PRITHIKA N** (**922519106117**) for **IBM-NALAIYATHIRAN** in **VII** semester of **B.E.**, degree course **in Electronics and Communication Engineering** branch during the academic year of 2022 – 2023.

Staff-In Charge

Sivalingam T

Evaluator

Dr P.S Gomathi

Head of the Department

Mrs. Dr P.S Gomathi

ACKNOWLEDGEMENT

First and foremost, we express my thanks to our parents for providing us a very nice environment for doing this mini project. We wish to express our sincere thanks to our founder and Chairman **Shri.V.S.BALSAMY** for his endeavour in educating us in this premier institution.

We wish to express our appreciation and gratefulness to our principal, **Dr.V.NIRMAL KANNAN** and vice principal **Mr.T.S.KIRUBASANKAR** for their encouragement and sincere guidance.

We are grateful to our head of the department **Mrs. Dr P.S GOMATHI** and our IBM Nalaiyathiran project coordinator **Mrs. Dr P.S GOMATHI** Department of Electronics and Communication Engineering for their valuable support.

We express our indebtedness to the supervisor of our IBM Nalaiyathiran project, **Mrs. Dr P.S GOMATHI** Professor, Department of Electronics and Communication Engineering, for guidance throughout the course of our project.

Our sincere thanks to all the teaching staff of V.S.B Engineering College and our friends for their help in the successful completion of this IBM Nalaiyathiran project work. Finally, we bow before God, the Almighty who always had a better plan for us. We give our praise and glory to Almighty God for successful completion of this IBM Nalaiyathiran.

ABSTRACT

Forest fires are occurring throughout the year with an increasing intensity in the caused by the actions of humans, butdifferent nature and environmental phenomena, like lightning strikes or spontaneous combustion of dried leaves or sawdust, can also be credited for their occurrence.

TABLE OF FIGURES

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

1.INTRODUCTION

Forest fires have been and still are serious problem for the European Union and for all other countries in Europe. In the year 2000, the EU has established the European Forest Fire Information system (EFFIS) , which will soon become part of the European Emergency Management Service, maintained by the CopernicusEarth Observation Program.

1.1 PROJECT OBJECTIVE

By the end of this project you will:

- You will be able to learn how to get and prepare the dataset.
- You will be able to know how to do image processing.
- You will understand how CNN layers are work.
- Classify images using a Convolutional Neural Network.
- You will be able to know what are the activation functions can be used.
- You will be able to know how to read images using OpenCV.
- You will know convolutional Neural Networks forComputer vision AI Problems. Upon learning and completing all the above objectives, we can obtain a model which predicts the forest fire.

2.LITERATURE SURVEY

2.1 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 program at: http://www.copernicus.eu

3] The 2018 Attica wildfires Wikipedia webpage available at:https://en.wikipedia.org/wiki/2018_Attica_wildfires

4] Official webpage of the ALTi Transition VTOL UAV, [online] Available:https://www.altiuas.com/transition/

2.2 PROBLEM STATEMENT DEFINITION

An Enormous disastrous fire that spread over a timberland or area of forest which prompts harm in Natural life, people, property and Climate. The significant Causes Are Lightning. Flashes from Rock falls. Volcanic Ejection or some other manual Start from the People deliberately which prompts the accompanying drawbacks: A backwoods fire sets up the potential for soil erosion to occur, Forest fires always bring death to life of humans and animals, Uncontrolled fires can cause localized air pollution, Homes can be destroyed without compensation

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

WILD LIFE

SAYS

- *Natural causes such as lightning which set trees on fire.*
- *Vehicles crashes, irresponsible campfires.*
- *Unextinguished cigarettes, urbanization.*
- *Climate changes.*

THINKS

- Carefully extinguish forest fires and disposal of hot charcoal.
- Drown all fires.
- Keep vehicles off of dry grasses.
- Call 911 to report fire.

DOES

- Don't dispose the flammable products and plastics in the forest.
- Avoid deforestation.
- Avoid firing, fireworks nearby forests.
- Frequently check whether and drought conditions.

FEELS

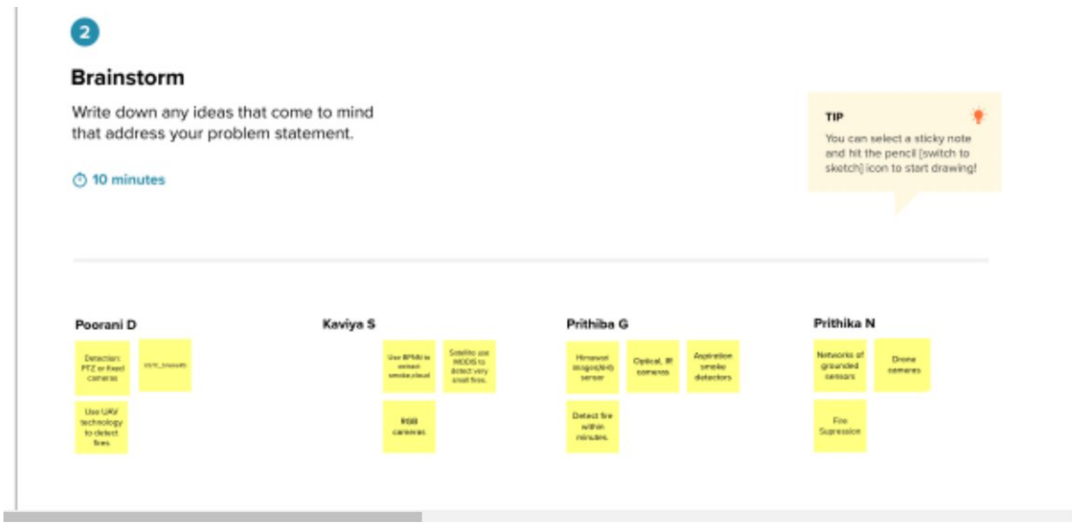
- Safe & Secure
- Easily Predictable
- Saves wildlife and tribal people

3.2 Ideation Phase & Brainstorming

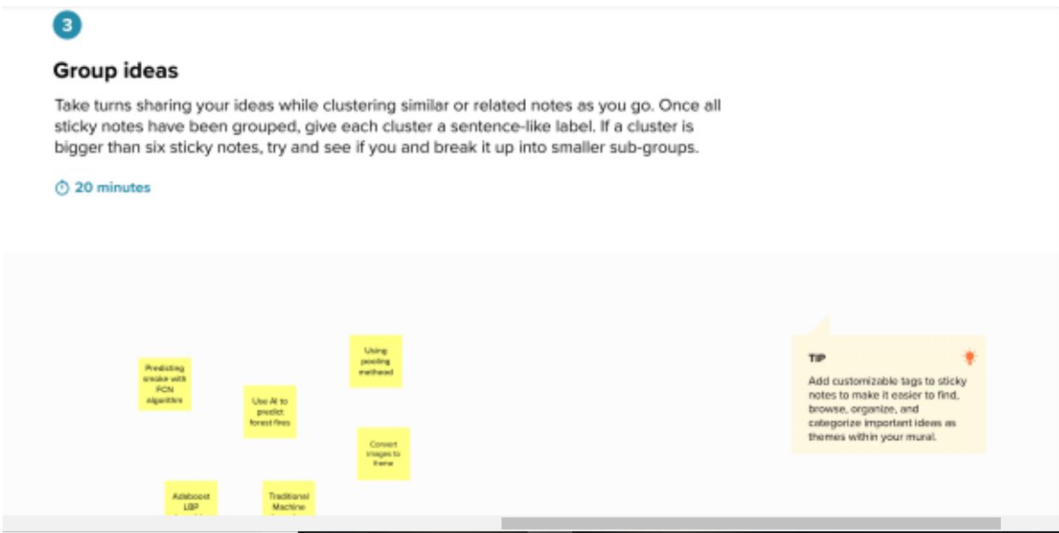
Brainstorm & Idea Prioritization Template:

Conceptualizing gives a free and open climate that supports everybody inside a group to partake in the imaginative reasoning cycle that prompts critical thinking. Focusing on volume over esteem, out-of-the-case thoughts are gladly received and based upon, and all members are urged to team up, helping each other foster a rich measure of clever fixes **.**

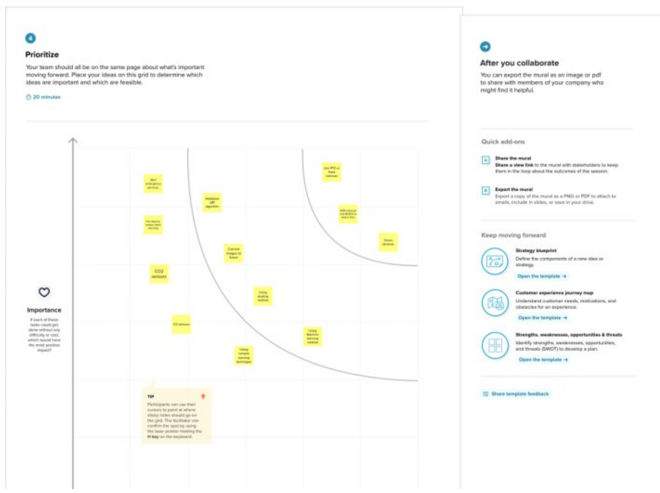
2. Brainstorm



Step-3: Group idea



Step-4: Prioritize



3.3 Proposed Solution

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S. No.	Parameter	Description
1	Problem Statement (Problem to besolved)	Forest fire prediction constitutes a significant component of forest fire management. It plays a major role in resource allocation, mitigation and recovery efforts. This paper presents a description and analysis of forest fire predictionmethods based on artificial intelligence. A novelforest fire risk prediction algorithm, based on support vector machines, is presented. The algorithm depends on previous weather conditions in order to predict the fire hazardlevel of a day.
2	Idea / Solution description	• Avoid burning wastes around dry grass. • Don't start a fire on a windy day. • Use a can or fire pit. • Never burn household wastes when any regulations of wildfire prevention policy prohibit it. • Don't throw explosives and combustibles into the fire.
3	Novelty / Uniqueness	Whenever you smoke, douse your butts with water and place them in a fire-proof containerto safely dispose of after you're sure they've gone out. And whatever you do, don't toss them on the ground. The device detects the high temperature , if theforest burns , the smoke will be absorbed and it prevents the forest.

4	Social Impact / Customer Satisfaction	Forest fires cause a loss of natural resources, depleting of soil biomass resulting in the loss ofvarious mobile nutrient
5	Business Model (Revenue Model)	• Drones • Robots • satellites.
6	Scalability of the Solution	Forest fire prediction constitutes a significant component of forest fire management. It plays a major role in resource allocation, mitigation and recovery efforts. This paper presents a description and analysis of forest fire predictionmethods based on artificial intelligence. A novelforest fire risk prediction algorithm, based on support vector machines, is presented. The algorithm depends on previous weather conditions in order to predict the fire hazard level of a day -The problem is done.

Project Design phase – I

Problem Solution fit

Project name: Team Id : PNT2022TMID33568

1.Customer segments:– In the early days for detecting the fire hazards by following some segments 1.Terrestrial Systems 2.Traditional Methods 3. Deep Learning Methods these are the segments which they used in their early days	**6.Customer constrains:–** The primary constraint on the fire detection system is to detect a developing fire prior to belt ignition, or as quickly as possible thereafter before the onset of rapid flame spread can begin	**5.Available solutions** Ionization smoke detectors (the most common in home use) detect the particles in smoke. As smoke passes through the chamber, the particles are ionized. These particles may then be detected by charged plates in the detector
2.Jobs to be done :- When the fire alarm system detects smoke, heat, or water movement, it alerts occupants of the building using both audible and visible alarms. These alarms will be bright, loud, obnoxious, and impossible to ignore, which help mobilize individuals to follow your evacuation plan.	**9.Problem route cause:-** Fire alarm systems are in place to do two major things; detect fire and alert occupants of the fire while giving them enough time to vacate the area. ... Dirty Smoke Detectors Dirt. Dust. Lint. Small Insects.	**7.Behavior:-** fire alarm system warns people when smoke, fire, carbon monoxide or other fire-related emergencies are detected. These alarms may be activated automatically from smoke detectors and heat detectors or may also be activated via manual fire alarm activation devices such as manual call points or pull stations.
3.Triggers:- Automatic initiating devices – automatic initiation devices trigger the fire alarm system automatically when a fire happens. These devices include heat, flame and smoke detection. When heat, flames or smoke is detected, the devices send a signal to a central control panel that activates the system	**10.Solution:-** Hence electronic circuits can be designed for the fire based alarms and they provide very high efficiency and can be used for the security reasons. Early fire detection is best achieved by the installation and maintenance of fire detection equipment in all rooms and areas of the house or building.	8.Channels of behavior:- Fire alarm control panel. Initiation devices. Pull stations. Smoke detectors. Duct detectors. Heat detectors. Beam detectors.

3.5 Solution Architecture

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridgesthe gap between business problems and technology solutions. Its goals are to:

- ❖ This planning illustrates the division of mass on UAVs and provides a betterunderstanding of the performance analysis of UAVs.
- ❖ Specifically, maximum take-off weights (MTOW) assess the UAV payload capacity atdifferent heights above the ground.
- ❖ The battery used on the UAV reserves the UAV in GPS-enabled environments for 107 minutes of duration, whereas on the GPS disabled environment, maximum flighttime is 87 minutes.

Example - Solution Architecture Diagram:

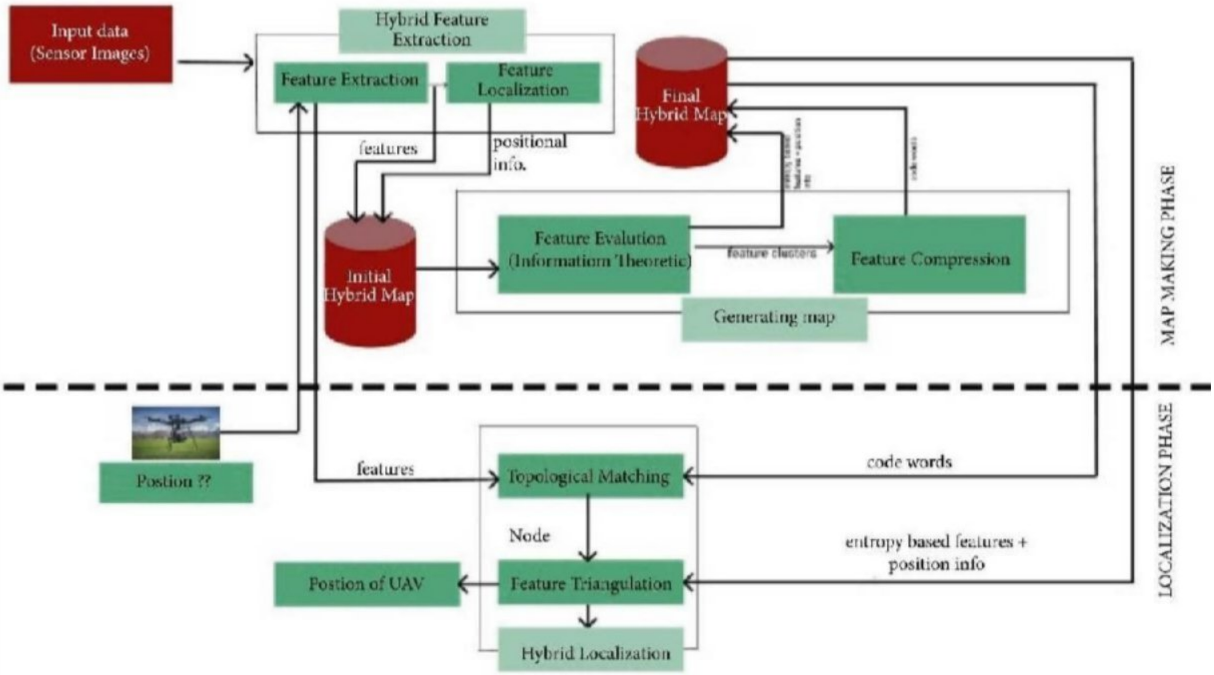
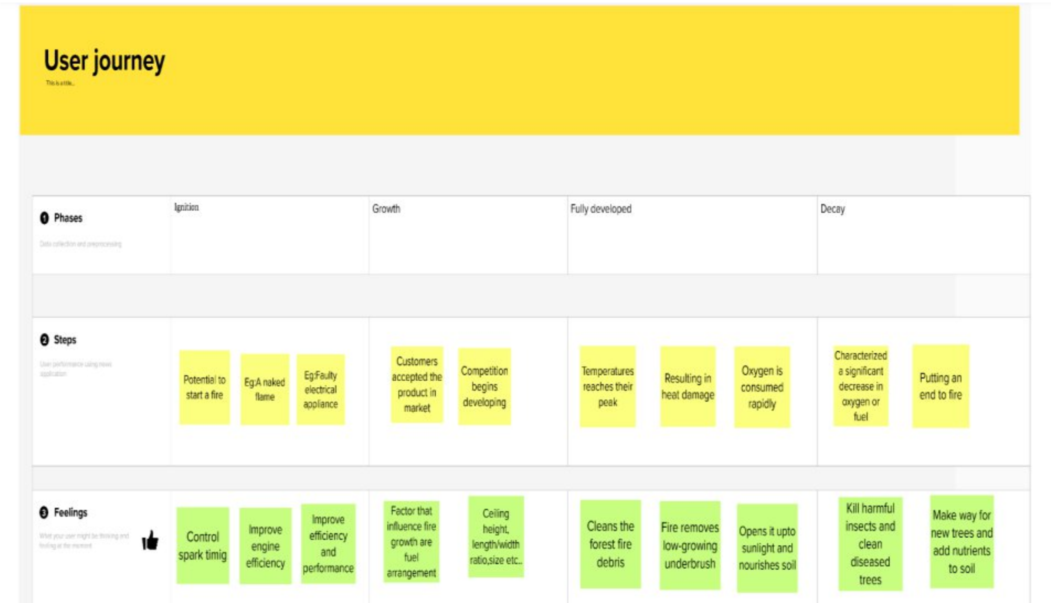


Figure 1: Architecture and data flow of the forest fire using UAV and learning techniques

Reference: https://www.hindawi.com/journals/cin/2022/3170244/

Customer journey map



4. REQUIREMENT ANALYSIS

4.1 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 Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Data Prediction	Scientists create computer models to predict wildfirepotential under a range of potential climate futures. Using different projections of temperature and precipitation, scientists predict where and when wildfires are most likely to occur.
FR-4	Using Sensors	This Bosch environment sensors installed in the forest fire detection system using artificial intelligence deployed as early wildfire warming tool.

4.2 NON-FUNCTIONAL REQUIREMENTS:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Monitoring of the potential risk areas and an early detection of fire can significantly shorten the reaction time and also reduce the potential damage as well as the cost of fire fighting.
NFR-2	**Security**	A fire alarm system warns people when smoke, fire,carbon monoxide or other fire-related or general notification emergencies are detected.

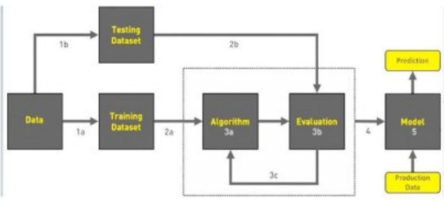
NFR-3	**Reliability**	It has achieved 1.24 seconds of classification timewith an accuracy of 91% and F1 score of 0.91.
NFR-4	**Performance**	initially, “internal” demonstration activities withoutuser involvement were organized, including controlled fires and artificial smoke tests to validate system functionalities, verify the correct operation of sensors and test system performance.
NFR-5	**Availability**	The experiment results show that the proposed h- EfficientDet can detect the fire in real-time with the detection speed of 21 FPS. The detection accuracy is up to 98.35% with a low miss detection rate.
NFR-6	**Scalability**	The current requirement for a cargo compartment detection system is that a fire has to be detected in 1 minute, and in that time be so small that the fire isnot a significant hazard to the airplane. Nuisance alarms also plague the industry, with upwards of 90% of fire alarms being false warnings.

5.PROJECT DESIGN

USER STORIES

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.



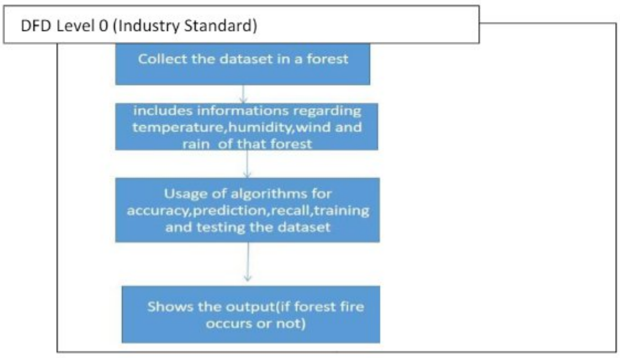
1 .Collect Data

2. Evaluate Data set

3. Implement Algorithm

4 Evaluate accuracy of algorithms

5 Display Results



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Environmentalist	Collect the data	USN-1	As an Environmentalist,it is necessary to collect the data of the forest which includes temperature,humidity,wind and rain of the forest	It is necessary to collect the right data else the prediction may become wrong	High	Sprint-1
		USN-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithms	Medium	Sprint-2
	Implement Algorithm	USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	High	Sprint-2
		USN-4	Evaluate the Dataset	Data is evaluated before processing	Medium	Sprint-1
	Evaluate Accuracy of Algorithm	USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the right output	High	Sprint-3
	Display Results	USN-6	Outputs from each algorithm are obtained	It is highly used to predict the effect and to take precautionary measures.	High	Sprint-4

Technical Architecture

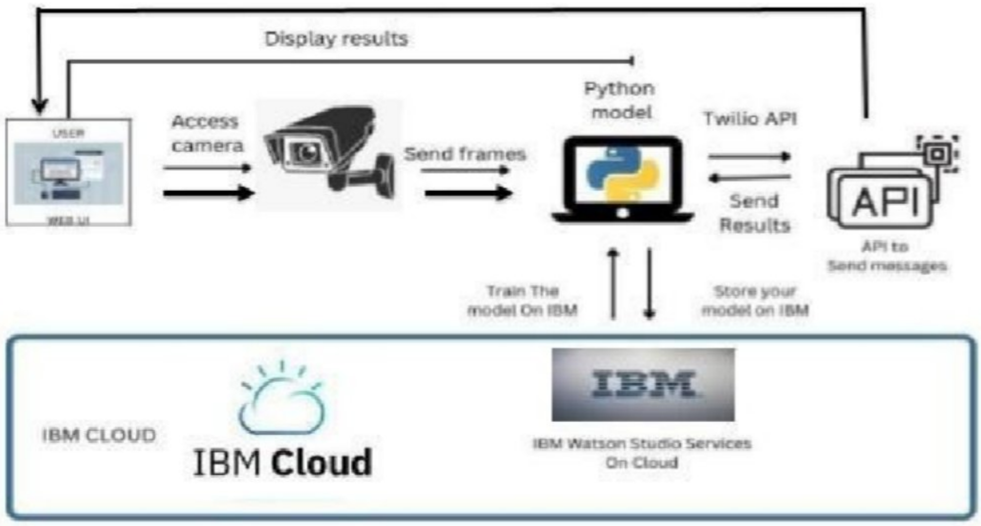


Table-1:
Components & Technologies:

Table-1:

Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	The user uses the console to access the interface	Python/HTML ,CSS , Javascript and react Js
2	Input	Video Feed	Web Camera/Video on a site
3	Conversion	Video inputted is converted into Frames	Frame Converter
4	Feeding the Model	The Frames are sent to the Deep learning model	Our Model
5	Dataset	Using Test set and Train set , train the model	Data set from Cloud Storage , Database
6	Cloud Database	The model is trained in the cloud more precise with detections more images can be added later on.	IBM Cloudant ,Python Flask.
7	Infrastructure (Server / Cloud), API	Application Deployment on Local System / Cloud Local ,Cloud Server Configuration , Twilio API to send messages	Java/python, React Js, JavaScript, HTML , CSS ,IBM Cloud ,OPEN CV, Anaconda Navigator ,Local.

Table-2:

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python Flask framework is used	Technology of Opensource framework
2.	Security Implementations	Mandatory Access Control (MAC) and Preventative Security Control is used	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	High scalability with 3-tier architecture	Web server – HTML ,CSS ,JavaScript Application server – Python , Anaconda Database server –IBM DB2
4.	Availability	Use of load balancing to distribute traffic across servers	IBM load balancer
5.	Performance	Enhance the performance by using IBM CDN	IBM Content Delivery Network

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collect the data.	USN-1	As an Environmentalist, it is necessary to collect the data of the forest which includes temperature, humidity, wind, and rain of the forest.	20	High	Poorani D Kaviya S
Sprint -2	Identifing algorithm which is used for detecting the forest fires.	USN-2	Identify algorithms that can be used for predicting the forest fires.	10	Medium	Prithiba G Prithika N
Sprint -3	Implement Algorithm	USN-3	Identify the accuracy of each algorithm in detecting the fires.	20	Medium	Poorani D Kaviya S
Sprint-4	Reliability	USN-4	Evaluate the data collected from algorithm with the defined data set.	20	Medium	Prithiba G Prithika N

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	5 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	05 Nov 2022

Sprint-3	20	8 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	10 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

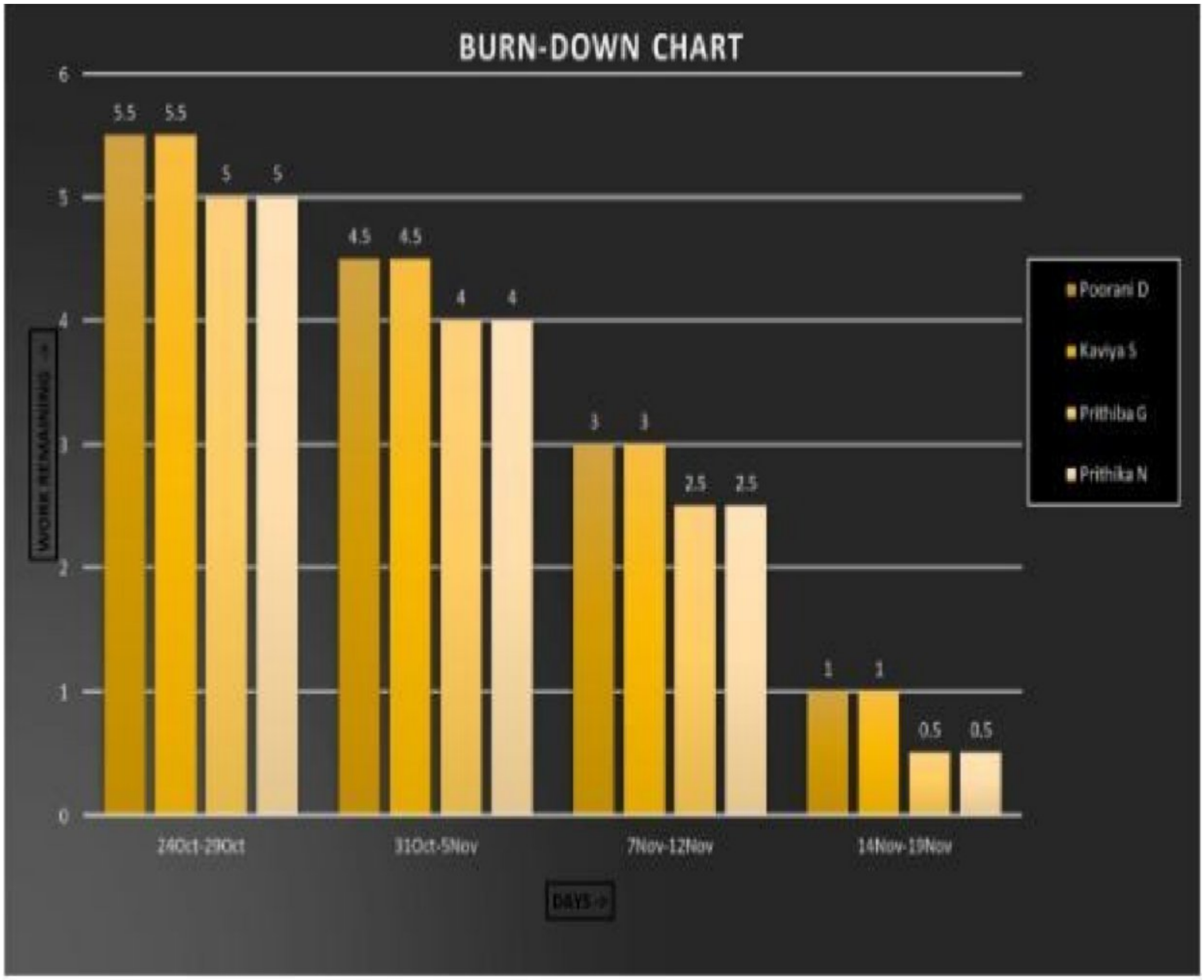
Sprint 1 average velocity:

Average velocity = 20 / 5 = 4.0

Sprint 2 average velocity:

Average velocity = 15 / 6 = 2.5

Sprint 3 average velocity:



BURN-DOWN CHART
WORK REMAINING →
DAYS →
24Oct-29Oct
31Oct-5Nov
7Nov-12Nov
14Nov-19Nov
Poorani D
Kaviya S
Prithiba G
Prithika N
5.5
5.5
5
5
4.5
4.5
4
4
3
3
2.5
2.5
1
1
0.5
0.5
0
1
2
3
4
5
6

6.PROJECT PLANNING AND SCHEDULING

DELIVERY OF SPRINT-1

Executable Program

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rot
ation_range=180,z oom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
x_train=train_datagen.flow_from_directory(r'C:\Users\USER\Document
s\Sem7\Naalaiyath ir
an\Dataset\Dataset\train_set', target_size=(128,128),
                        batch_size=
                        32,
                        class_mode
                        ='binary')
x_test=train_datagen.flow_from_directory(r'C:\Users\USER\Documents\
Sem7\Naalaiyathira n\Dataset\Dataset\test_set', target_size=(128,128),
                        batch_size=
                        32,
                        class_mode
                        ='binary')
x_train.class_indices from
tensorflow.keras.models import
Sequentialfrom
tensorflow.keras.layers
import Dense
from tensorflow.keras.layers import
Convolution2D,MaxPooling2D,Flatten     import
            warnings
            warnings.filterwarnings('ignore')
model=Sequential()
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()
```

```
In [1]: from tensorflow.keras.preprocessing.image import ImageDataGenerator

C:\anaconda\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this ver
sion of SciPy (detected version 1.23.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"

In [2]: train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)

In [3]: test_datagen=ImageDataGenerator(rescale=1./255)

In [5]: x_train=train_datagen.flow_from_directory(r'C:\Users\USER\Documents\Sem7\Naalaiyathiran\Dataset\Dataset\train_set', target_size=(
                                          batch_size=32,
                                          class_mode='binary')

Found 436 images belonging to 2 classes.

In [6]: x_test=train_datagen.flow_from_directory(r'C:\Users\USER\Documents\Sem7\Naalaiyathiran\Dataset\Dataset\test_set', target_size=(12
                                          batch_size=32,
                                          class_mode='binary')

Found 121 images belonging to 2 classes.

In [7]: x_train.class_indices

Out[7]: {'forest': 0, 'with fire': 1}

In [8]: from tensorflow.keras.models import Sequential
```

```
In [8]: from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Dense

In [9]: from tensorflow.keras.layers import Convolution2D,MaxPooling2D, Flatten

In [10]: import warnings
         warnings.filterwarnings('ignore')

In [11]: model=Sequential()

In [13]: model.add(MaxPooling2D(pool_size=(2,2)))

In [14]: model.add(Flatten())

In [70]: model.summary()

Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_4 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 flatten_4 (Flatten)         (None, 127008)            0

=================================================================
Total params: 896
```

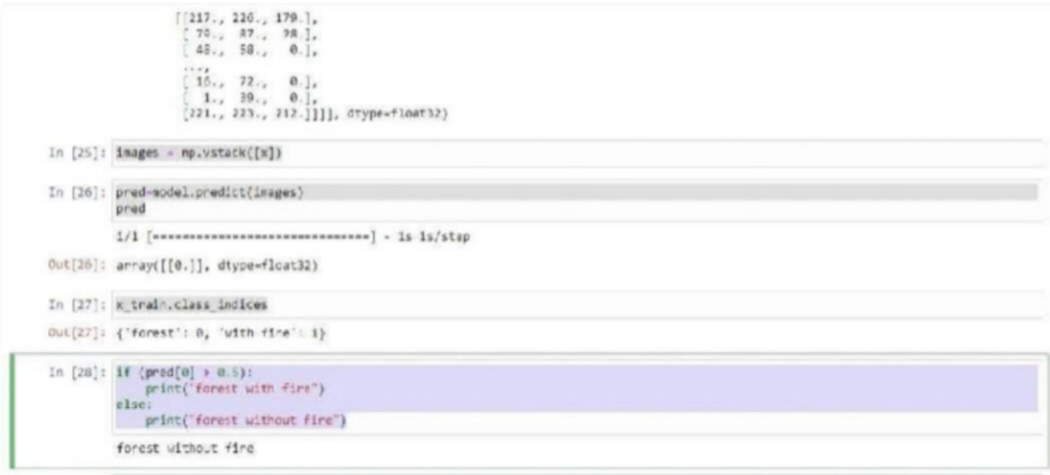
PROJECT DEVOLPMENT PHASE
DELIVERY OF SPRINT-2

Executable ProgramModel

Building:

```
model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['
accuracy'
]
) len(x_train) len(x_test)
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=1
0, validation_data=x_test,validation_steps=len(x_test)) import
tensorflow as tf from keras.models import load_model from
tensorflow.keras.preprocessing import imageimport numpy as np
import cv2 model.save('forestfire.h5')
model=load_model('forestfire.h5')testImg =
image.load_img(r'C:\Users\win\Desktop\Project_NT\test_set\forest\_
101542074_ g ettyimages_956391468.jpg')
testImgarrayImg =
image.img_to_array(testImg) arrayImg
x = np.expand_dims(arrayImg , axis =
0)X images =np.vstack([x])
pred=model.predict(images) Pred
x_train.class_indicesif (pred[0] > 0.5):
print("forest with fire")else:
print("forest without fire")
```

```
Epoch 1/10
14/14 [==============================] - 46s 3s/step - loss: 3.7642 - accuracy: 0.5550 - val_loss: 0.9342 - val_accuracy: 0.595
0
Epoch 2/10
14/14 [==============================] - 21s 2s/step - loss: 0.4257 - accuracy: 0.8050 - val_loss: 0.1760 - val_accuracy: 0.925
6
Epoch 3/10
14/14 [==============================] - 22s 2s/step - loss: 0.2191 - accuracy: 0.9083 - val_loss: 0.1141 - val_accuracy: 0.958
7
Epoch 4/10
14/14 [==============================] - 22s 2s/step - loss: 0.2520 - accuracy: 0.8991 - val_loss: 0.1058 - val_accuracy: 0.975
2
Epoch 5/10
14/14 [==============================] - 22s 2s/step - loss: 0.2192 - accuracy: 0.9014 - val_loss: 0.1065 - val_accuracy: 0.966
9
Epoch 6/10
14/14 [==============================] - 22s 2s/step - loss: 0.1942 - accuracy: 0.9106 - val_loss: 0.0938 - val_accuracy: 0.975
2
Epoch 7/10
14/14 [==============================] - 21s 2s/step - loss: 0.1684 - accuracy: 0.9358 - val_loss: 0.1383 - val_accuracy: 0.942
1
Epoch 8/10
14/14 [==============================] - 22s 2s/step - loss: 0.1872 - accuracy: 0.9266 - val_loss: 0.1577 - val_accuracy: 0.900
8
Epoch 9/10
14/14 [==============================] - 25s 2s/step - loss: 0.1643 - accuracy: 0.9312 - val_loss: 0.0874 - val_accuracy: 0.983
5
Epoch 10/10
14/14 [==============================] - 62s 5s/step - loss: 0.1640 - accuracy: 0.9220 - val_loss: 0.0809 - val_accuracy: 0.975
2
```



```
       [[217., 226., 179.],
        [ 70.,  87.,  78.],
        [ 48.,  58.,   0.],
        ...,
        [ 16.,  72.,   0.],
        [  1.,  39.,   0.],
        [221., 223., 212.]]]], dtype=float32)

In [25]: images = np.vstack([x])

In [26]: pred=model.predict(images)
         pred

         1/1 [==============================] - 1s 1s/step

Out[26]: array([[0.]], dtype=float32)

In [27]: x_train.class_indices

Out[27]: {'forest': 0, 'with fire': 1}

In [28]: if (pred[0] > 0.5):
             print("forest with fire")
         else:
             print("forest without fire")

         forest without fire
```

PROJECT DEVELOPMENT PHASE Delivery of Sprint -3

Executable

ProgramVideo Analysis:

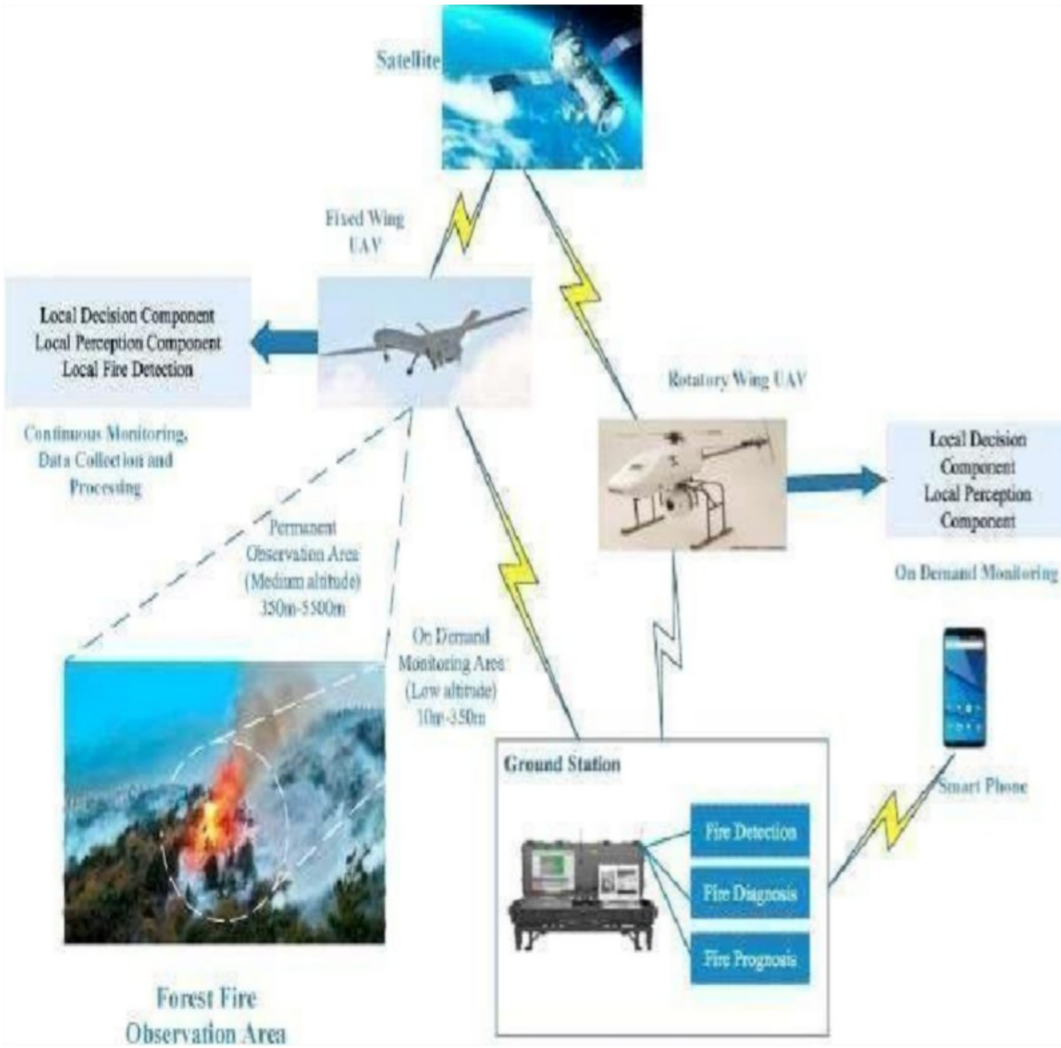
import cv2 import numpy as np from keras.preprocessing import imagefrom keras.models import load_model from twilio.rest importClient
!pip install twilio
model=load_model('forestfire.h5')
video=cv2.VideoCapture(r'C:\Users\win\Desktop\Project_NT\video.mp 4') name=['forest','with fire']
#predict=model.predi ct(x)import keras from tensorflow.keras.utils import load_img, img_to_arraywhile(1):
success,frame=video.read()
cv2.imwrite("image.jpg",frame)
img=keras.utils.load_img("image.jpg") img=cv2.resize(frame, (128,128))
x=keras.utils.img_to_array(img)
x=np.expand_dims(x,axis=0)
dim=(128,128)

```
#
        x=x.reshape(1
28,128,
        3) #
        x=cv2.resize(x
, (128,128))  pred =
model.predict(x)

#pred=model.predict_c
lass es(x)p=pred[0]
print(pred)
#                                          cv2.putText(frame,"predicted
class="+str(name[p]),(100,100),cv2.FONT_HERSHEY_SIMPL
                                                  EX,1,(0,0,0),1
) if pred[0]==1:
  account_sid='AC63518ea0e5f8e919ee2a4dc4dc17cdb
  6'auth_token='e5413a0fd6c65647ca88e8cb0cd33fac'
  client=Client(account_sid,auth_token)

  message=client.messages.create(body='Forest Fire is
  detected,stayalert',from_='+1          989  762
                      1639',        to='+91 9344394743')
  print(message.si d) print('Fire Detected')
print('SMS sent!')else:
  print("No Danger")
cv2.imshow("image",
frame)
#if cv2.waitkey(1) & 0xFF==
ord('a'):#breakvideo.release()
cv2.destroyAllWindows()
```

```
In [37]: #predict=model.predict(x)
if pred[0]==0:
    account_sid='AC63518ea0e5f8e919ee2a4dc4dc17cdb6'
    auth_token='e5413a0fd6c65647ca88e8cb0cd33fac'
    client=Client(account sid,auth token)
    message=client.messages.create(body='Forest Fire is detected,stay alert',
    from_='+1 989 762 1639',
    to='+91 9344394743')
    print(message.sid)
    print('Fire Detected')
    print('SMS sent!')
else:
    print("No Danger")
cv2.imshow("image",frame)
#if cv2.waitkey(1) & 0xFF== ord('a'):
    #break
video.release()
cv2.destroyAllWindows()

SMb6dd721ea1b8b7da9ef4ced9d83c3294
Fire Detected
SMS sent!
```

PROJECT DEVELOPMENT PHASE
DELIVERY OF SPRINT -4

Due to low spatial resolution and low temporal resolution, satellite images cannot beused for wildfire detection. However, UAV gives high-resolution images and is cost- effective for wildfire detection. Fig. 11 gives a proposed architecture of forest fire detection and monitoring system based on UAV. The basic elements of the UAV-based forest fire system cover functions of monitoring (gives the potential of fire), detection (gives a triggering alarm), diagnosis and prognosis, which are initialized by fire operators after getting the triggering alarm. The main function of the diagnosis is to find detailed information regarding the location and extent of the fire. And also track its evolution whereasin prognosis it tracks and predicts the evolution of future fire in real-time using information (wind and fire conditions) provided by onboard remote monitoring sensors that are installed on UAV. These functions can be carried out with the help of single UAV, or by a group of UAVs (with multiple sensors having different functions) along with the central ground station. The main aim of using UAV is to predict the occurrence of fire, track its location, and to give real-time information to fire-fighters or execute its operation with UAV. To acquire these goals UAV-based system includes (i)



various sensors, GPS receivers, cameras, Inertial Measurement Units (IMUs), (ii) algorithms and strategies, (iii) autonomous Guidance Navigation and Control (GNC), (iv)localization, deployment and control system, and (v) ground station.

{
"nbformat": 4,
"nbformat_minor
": 0,"metadata":
{
"colab":
{ "provenanc
e": [],
"collapsed_sections": []
},
"kernelspec":
{ "name":
"python3",
"display_name": "Python 3"
},
"language_inf
o": { "name":
"python"
}
},
"cells": [
{
"cell_type":
"markdown",
"source": [
"#Importing Keras libraries"
],
"metadata": {
"id": "5UKKDaCfUNCG"
}
},
{
"cell_type":
"code",
"source":
["import
keras"
],
"metadata": {
"id": "ipCWLeFzUXGa"
},
"execution_count
": 2,"outputs": []

},
{
"cell_type":
"markdown",
"source": [
"#Importing ImageDataGenerator from Keras"
]"metadata": {
"id": "RNeZ0vrEUc55"
}
},
{
"cell_type":
"code",
"source": [
"from keras.preprocessing.image import ImageDataGenerator"
],
"metadata": {
"id": "Mx2JxqSDUnPk"
},
"execution_count
": 3,"outputs": []
},
{
"cell_type":
"markdown",
"source": [
"#Defining the Parameters"
],
"metadata": {
"id": "p7bB9Cd1b8kG"
}
},
{
"cell_type":
"code",
"source": [
"from google.colab import
drive\n",
"drive.mount('/content/drive
')"
],
"metadata": {

"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "jkgacY2Su-U1",
"outputId": "a2210a0f-08b2-40fa-e726-3e729fb9710f"
},
"execution_count
": 4,"outputs": [
{
"output_type":
"stream","name":
"stdout","text": [
"Mounted at /content/drive\n"
]
}
]
},
{
"cell_type":
"code",
"source": [
"cd/content/drive/MyDrive/IBM PROJECT"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "Dcr_Un4D7On8",
"outputId": "6b6828b3-f29a-4286-e02f-70e7ff5a45f7"
},
"execution_count
": 5,"outputs": [
{
"output_type":
"stream","name":
"stdout",
"text": [
"/content/drive/MyDrive/IBM PROJECT\n"
]
}
]
},

{
"cell_type":
"code",
"source": [
"!unzip 'dataset.zip'"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "ba6mSnqc7W8Z",
"outputId": "6d098827-a0ac-46fa-b9dc-cc9f02b1c3c0"
},
"execution_count": 22,
"outputs": [
{
"output_type":
"stream","name":
"stdout",
"text": [
"Archive: dataset.zip\n",
" extracting: dataset/null
\n"
," creating:
dataset/Test_set/\n",
" creating: dataset/Test_set/Forest with fire/\n",
" inflating: dataset/Test_set/Forest with
fire/180802_CarrFire_010_large_700x467.jpg \n"," inflating:
dataset/Test_set/Forest with fire/19464620_401.jpg \n",
" inflating: dataset/Test_set/Forest with
fire/599857.jpg \n"," inflating:
dataset/Test_set/Forest with
fire/b2683a16be44f230a90f50bb944315e994b43042e43df798bb4feaf2
7e0f8bc8.jpg \n"," inflating: dataset/Test_set/Forest with
fire/Bandipur_fires_2019.jpg \n",
" inflating: dataset/Test_set/Forest with
fire/climate_change_forest_fire.jpeg \n"," inflating:
dataset/Test_set/Forest with fire/cumming_forest_fires.jpg \n",
" inflating: dataset/Test_set/Forest with
fire/deerfire_high_res_edit.jpg \n"," inflating:
dataset/Test_set/Forest with fire/defense_large.jpg \n",

" inflating: dataset/Test_set/Forest with fire/f90_fire_1168x657.jpg \n",
" inflating: dataset/Test_set/Forest with fire/file_20190308_155532_1nvts27.jpg \n"," inflating: dataset/Test_set/Forest with fire/Fire_2_696x392.jpg \n",
" inflating: dataset/Test_set/Forest with fire/Forest_Fire (3).jpg \n", " inflating: dataset/Test_set/Forest with fire/Forest_Fire (4).jpg \n", " inflating: dataset/Test_set/Forest with fire/forest_fire_0.jpg \n",
" inflating: dataset/Test_set/Forest with fire/forest_fire_2268729_1280.jpg \n", " inflating: dataset/Test_set/Forest with fire/forest_fire_at_bandipur_tiger_reserve_04f0dd5e_3919_11e9_9798_f85edb62de22.jpg \n", " inflating: dataset/Test_set/Forest with fire/Forest_fire_MNRF_esize_IMG_6743.jpg \n", " inflating: dataset/Test_set/Forest with fire/FORESTFIRE (1).jpg \n",
" inflating: dataset/Test_set/Forest with fire/forestfire.jpg \n",
" inflating: dataset/Test_set/Forest with fire/forestfire_kApH 621x414_LiveMint.jpg \n"," inflating: dataset/Test_set/Forest with fire/GettyImages_482867948.0.jpg \n",
" inflating: dataset/Test_set/Forest with fire/GettyImages_862092472_1160x773.jpg \n"," inflating: dataset/Test_set/Forest with fire/growing_risk_future_wildfires_adobestock_199370851.jpeg \n",
" inflating: dataset/Test_set/Forest with fire/header_hypermaritime_fire.jpg \n"," inflating: dataset/Test_set/Forest with fire/horseshoe_bay_fire.jpg \n",
" inflating: dataset/Test_set/Forest with fire/How_to_Protect_Your_Home_From_Forest_Fire_1024x588.jpg \n",
" inflating: dataset/Test_set/Forest with fire/image.jpg \n",
" inflating: dataset/Test_set/Forest with fire/in_forestfire1_(1).jpg \n",
" inflating: dataset/Test_set/Forest with fire/in_forestfires_shutter.jpg \n", " inflating: dataset/Test_set/Forest with fire/in_forestfiress_(1).jpg \n",
" inflating: dataset/Test_set/Forest with fire/korea_forest_fire_4.jpg \n", " inflating: dataset/Test_set/Forest with fire/louisiana_forest_fire.jpg \n", " inflating: dataset/Test_set/Forest with fire/maxresdefault.jpg \n",
" extracting: dataset/Test_set/Forest with fire/null \n",
" inflating: dataset/Test_set/Forest with

fire/photo_1555842027_660d9d8df55b.jpg \n", " inflating: dataset/Test_set/Forest with fire/RED7_May29_1986.jpg \n",
" inflating: dataset/Test_set/Forest with fire/ring_of_fire_bailey_colorado_rocky_mountain_forest_wildfire_picture_id157384116.jpg \n",
" inflating: dataset/Test_set/Forest with fire/rsz_1fire_pix_accumulation_640_1_2.jpg \n"," inflating: dataset/Test_set/Forest with fire/shutterstock_48032821.jpg \n",
" inflating: dataset/Test_set/Forest with fire/shutterstock_710588224.jpg \n",
" inflating: dataset/Test_set/Forest with fire/skynews_ashdown_forest_fire_4653702.jpg \n"," inflating: dataset/Test_set/Forest with fire/The_Rim_Fire_in_the_Stanislaus_National_Forest_near_in_California_began_on_Aug._17_2013_0004.jpg \n",
" inflating: dataset/Test_set/Forest with fire/Untitled_design_161_866x487.jpg \n"," inflating: dataset/Test_set/Forest with fire/Uttarakhand_forest_fire.jpeg \n",
" inflating: dataset/Test_set/Forest with fire/uttarakhand_forest_fire_750x500.jpg \n"," inflating: dataset/Test_set/Forest with fire/Washington_DNR_Chiwaukum_WildFires_2014_2015_WA_DNR_cropped.jpg \n"," inflating: dataset/Test_set/Forest with fire/Wild_fires.jpg \n",
" inflating: dataset/Test_set/Forest with fire/wildfire_header_image_1024x1024.jpg \n"," inflating: dataset/Test_set/Forest with fire/X80IR0GYSsG6RaiLcsXX.jpg \n",
" creating: dataset/Test_set/Forest without fire/\n",
" inflating: dataset/Test_set/Forest without fire/_101542074_gettyimages_956391468.jpg \n", " inflating: dataset/Test_set/Forest without fire/0.48007200_1530881924_final_forest.jpg \n", " inflating: dataset/Test_set/Forest without fire/0.64133000_1519374442_forest_deep.jpg \n", " inflating: dataset/Test_set/Forest without fire/0.72918000_1559733279_forests1_gettyimages_.jpg \n",
" inflating: dataset/Test_set/Forest without fire/0.98884800_1554454572_spin.jpg \n",
" inflating: dataset/Test_set/Forest without fire/01_NeilBurnell_Mystical_photoverticall.jpg \n"," inflating: dataset/Test_set/Forest without fire/091318_LH_forest_loss_main_FREE.jpg \n",
" inflating: dataset/Test_set/Forest without

fire/1_chimp.jpg \n"," inflating:
dataset/Test_set/Forest without fire/1009821.jpg
\n",
" inflating: dataset/Test_set/Forest without
fire/111188170_river_in_the_mountain_forest.jpg\n",
" inflating: dataset/Test_set/Forest without fire/1170x500_Ireland_web.jpg \n", "
inflating: dataset/Test_set/Forest without fire/1200px_Mountainarea.jpg \n", "
inflating: dataset/Test_set/Forest without fire/146019.jpg \n",
" inflating: dataset/Test_set/Forest without fire/1506697583544.jpg \n"," inflating:
dataset/Test_set/Forest without fire/1551622076_img_5241.jpg \n", " inflating:
dataset/Test_set/Forest without fire/1551903255_2942.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/1556508284pexels_photo_1179229.jpeg \n"," inflating:
dataset/Test_set/Forest without
fire/16475617_web1_ForestHealth_VIB_190418_2.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/18435324436_6a20e7fc70_k.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/200px_View_of_Lake_Moraine.jpg \n"," inflating:
dataset/Test_set/Forest without fire/2017_10_12_09_01_56.jpg \n",
" inflating: dataset/Test_set/Forest without fire/220px_The_forest_near_Blatets
Vinitsa.JPG
\n"
, " inflating: dataset/Test_set/Forest without
fire/23769_pcbkrdossw_1500310327.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/2E215059_55C7_417B_8DDD693AA6E416B4_source.jpg \n","
inflating: dataset/Test_set/Forest without
fire/31105585103_4c32392ac1_k.jpg \n"," inflating:
dataset/Test_set/Forest without fire/381417.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/4632581118_dcfb9ed978_b.jpg.860x0_q70_crop_scale.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/4fe3f4e45d6bb93070989b8bbb77d30e.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/4k_timelapse_heavy_rain_on_mountain_forest_tree_fog_motion_bad_weather_
condition_rfift1zh
thumbnail.png \n",
" inflating: dataset/Test_set/Forest without
fire/5325.jpg \n"," inflating:
dataset/Test_set/Forest without fire/5472.jpg \n",
" inflating: dataset/Test_set/Forest without

fire/559672101517195076987621071193712n.jpg
\n",
" inflating: dataset/Test_set/Forest without
fire/7_18_MedicineBowHikingFishing_KL_131
9_4003.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/78closer_exterior_view_of_northern_Rocky_Mountain
_forest.jpg \n"," inflating: dataset/Test_set/Forest
without
fire/79806645_landscape_of_tall_grass_field_green_mountain_forest.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/7ece9530_9cf1_11e9_bf3c_4fe6e6438142.jpg\n",
" inflating: dataset/Test_set/Forest without fire/852554.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/91ErlodmE5L.jpg \n"," inflating: dataset/Test_set/Forest
without fire/91sALlLjdLL.jpg \n",
" inflating: dataset/Test_set/Forest without fire/a.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/A003_C014_0724EC.0000000.jpg \n", " inflating:
dataset/Test_set/Forest without fire/acadia_np_622419_High_Res.jpg \n",
" inflating: dataset/Test_set/Forest
withoutfire/Adirondack_Habitats_Mixed_Forest_Barnum_Brook_Trail_P
aul_Smiths_VIC_23_May_2012_61.jpg
\n",
" inflating: dataset/Test_set/Forest without
fire/Adirondack_Habitats_Mixed_Forest_Jenkins_Mountain_Trail_Paul_Smiths_VIC
_12_August_2013_7 1.jpg \n",
" inflating: dataset/Test_set/Forest without fire/Almaty_Mountains.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/altai_regio_pine_forest_1550x804.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/amazing_green_trees_in_dense_mountain_forest_nature_landscape_in_4k_aerial
_view_b7oqfwut g_thumbnail_.png \n",
" inflating: dataset/Test_set/Forest without fire/aokigaharaforest_1_star.jpg
\n",
" inflating: dataset/Test_set/Forest without
fire/appalachian_mtns_forest_usda_srs.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/austria_mountain_forest_landscape_wallpaper_1920x1200.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/bavarian_forest_3385966_1280_1.jpg \n"," inflating:
dataset/Test_set/Forest without

fire/beautiful_green_forest_range_images_picture_photos_wallpaper_d
ownload.jpg \n"," inflating: dataset/Test_set/Forest without
fire/beautiful_morning_mountain_forest_scenery_free_stock_photos_picjumbo
_DSC02176.jpg \n"," inflating: dataset/Test_set/Forest without
fire/beautiful_mountain_forest_wallpaper_1920x1200.jpg \n",
" inflating: dataset/Test_set/Forest without fire/beech_oak_forest_560.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/britishforest_696483208.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/canva_waterfalls_in_the_mountain_forest_MADR_l3e0W4.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/ce8b73d56bd242eb1f2369fe856f1c63_700.jpg
\n",
" inflating: dataset/Test_set/Forest without
fire/cold_daylight_environment_1423600_640x427.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/Deciduous_forest.jpg \n"," inflating:
dataset/Test_set/Forest without fire/DSC_0072_Forest.jpg
\n",
" inflating: dataset/Test_set/Forest without fire/DSCN8209(1)_250x333.JPG
\n",
" inflating: dataset/Test_set/Forest without
fire/Dupont_Forest_Triple_Falls_Rob_Travis_2_1.jp\n",
" inflating: dataset/Test_set/Forest without fire/easternArc.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/em_tiger_mountain_570x284_nc.jpg \n"," inflating:
dataset/Test_set/Forest without fire/europeslostf.jpg \n","
inflating: dataset/Test_set/Forest
withoutfire/Fall_Trees_Alaskan_Adventure_by_John_Simeone.jpg \n",
" inflating: dataset/Test_set/Forest without
fire/Fallen_e1499694482182.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/fanas_gregorschmucki_2016_28_1_2600x1733.jpg \n"," inflating:
dataset/Test_set/Forest without
fire/fantastic_dark_landscape_of_mountain_forest_in_clouds_fog_or_mist_picture_i
d688948674.jpg
\n",\n",
" inflating: dataset/Test_set/Forest without
fire/fb26e4420f3ae3b12788fffdea0fb132_full.jpg

" inflating: dataset/Test_set/Forest without
fire/file_20180627_112634_8tdgn4.jpg \n"," extracting:
dataset/Test_set/Forest without fire/null \n",

" extracting: dataset/Test_set/null
\n"," creating: dataset/Train_set/\n",
" inflating: dataset/Train_set/with fire (1).png
\n", " inflating: dataset/Train_set/with fire
(10).jpg \n",
" inflating: dataset/Train_set/with fire (100).jpg
\n"," inflating: dataset/Train_set/with fire
(101).jpg \n"," inflating:
dataset/Train_set/with fire (102).jpg \n","
inflating: dataset/Train_set/with fire (103).jpg
\n"," inflating: dataset/Train_set/with fire
(104).jpg \n"," inflating:
dataset/Train_set/with fire (105).jpg \n","
inflating: dataset/Train_set/with fire (106).jpg
\n"," inflating: dataset/Train_set/with fire
(107).jpg \n"," inflating:
dataset/Train_set/with fire (108).jpg \n","
inflating: dataset/Train_set/with fire (109).jpg
\n"," inflating: dataset/Train_set/with fire
(11).jpg \n", " inflating: dataset/Train_set/with
fire (110).jpg \n"," inflating:
dataset/Train_set/with fire (111).jpg \n","
inflating: dataset/Train_set/with fire (112).jpg
\n"," inflating: dataset/Train_set/with fire
(113).jpg \n","
inflating:dataset/Train_set/with fire
(114).jpg \n"," inflating:
dataset/Train_set/with fire (115).jpg
\n"," inflating: dataset/Train_set/with fire
(116).jpg \n"," inflating:
dataset/Train_set/with fire (117).jpg \n","
inflating: dataset/Train_set/with fire (118).jpg
\n"," inflating: dataset/Train_set/with fire
(119).jpg \n"," inflating:
dataset/Train_set/with fire (12).jpg \n", "
inflating: dataset/Train_set/with fire (120).jpg
\n"," inflating: dataset/Train_set/with fire
(121).jpg \n"," inflating:
dataset/Train_set/with fire (122).jpg \n","
inflating: dataset/Train_set/with fire (123).jpg
\n"," inflating: dataset/Train_set/with fire

(124).jpg \n"," inflating:
dataset/Train_set/with fire (125).jpg \n","
inflating: dataset/Train_set/with fire (126).jpg
\n"," inflating: dataset/Train_set/with fire
(127).jpg \n"," inflating:
dataset/Train_set/with fire (128).jpg \n","
inflating: dataset/Train_set/with fire (129).jpg
\n"," inflating: dataset/Train_set/with fire
(13).jpg \n", " inflating: dataset/Train_set/with
fire (130).jpg \n"," inflating:
dataset/Train_set/with fire (131).jpg \n","
inflating: dataset/Train_set/with fire (132).jpg
\n"," inflating: dataset/Train_set/with fire
(133).jpg \n"," inflating:
dataset/Train_set/with fire (134).jpg \n","
inflating: dataset/Train_set/with fire (135).jpg
\n"," inflating: dataset/Train_set/with fire
(136).jpg \n"," inflating:
dataset/Train_set/with fire (137).jpg \n","
inflating: dataset/Train_set/with fire (138).jpg
\n"," inflating: dataset/Train_set/with fire
(139).jpg \n"," inflating:
dataset/Train_set/with fire (14).jpg \n", "
inflating: dataset/Train_set/with fire (140).jpg
\n"," inflating: dataset/Train_set/with fire
(141).jpg \n"," inflating:
dataset/Train_set/with fire (142).jpg \n","
inflating: dataset/Train_set/with fire (143).jpg
\n"," inflating: dataset/Train_set/with fire
(144).jpg \n"," inflating:
dataset/Train_set/with fire (145).jpg \n","
inflating: dataset/Train_set/with fire (146).jpg
\n"," inflating: dataset/Train_set/with fire
(147).jpg \n"," inflating:
dataset/Train_set/with fire (148).jpg \n","
inflating: dataset/Train_set/with fire (149).jpg
\n"," inflating: dataset/Train_set/with fire
(15).jpg \n", " inflating: dataset/Train_set/with
fire (16).jpg \n", " inflating:
dataset/Train_set/with fire (17).jpg \n", "
inflating: dataset/Train_set/with fire (18).jpg

\n", " inflating: dataset/Train_set/with fire (19).jpg \n", " inflating: dataset/Train_set/with fire (2).jpeg \n", " inflating: dataset/Train_set/with fire (2).jpg \n",

" inflating: dataset/Train_set/with fire (2).png \n",

" inflating: dataset/Train_set/with fire (20).jpg \n", " inflating: dataset/Train_set/with fire (21).jpg \n", " inflating: dataset/Train_set/with fire (22).jpg \n", " inflating: dataset/Train_set/with fire (23).jpg \n", " inflating: dataset/Train_set/with fire (24).jpg \n", " inflating: dataset/Train_set/with fire (25).jpg \n", " inflating: dataset/Train_set/with fire (26).jpg \n", " inflating: dataset/Train_set/with fire (27).jpg \n", " inflating: dataset/Train_set/with fire (28).jpg \n", " inflating: dataset/Train_set/with fire (29).jpg \n", " inflating: dataset/Train_set/with fire (3).jpeg \n"," inflating: dataset/Train_set/with fire (3).jpg \n", " inflating: dataset/Train_set/with fire (3).png \n", " inflating: dataset/Train_set/with fire (30).jpg \n"," inflating: dataset/Train_set/with fire (31).jpg \n"," inflating: dataset/Train_set/with fire (32).jpg \n"," inflating: dataset/Train_set/with fire (33).jpg \n"," inflating: dataset/Train_set/with fire (34).jpg \n"," inflating: dataset/Train_set/with fire (35).jpg \n"," inflating: dataset/Train_set/with fire (36).jpg \n"," inflating: dataset/Train_set/with fire (37).jpg \n"," inflating: dataset/Train_set/with fire (38).jpg \n"," inflating: dataset/Train_set/with fire (39).jpg \n"," inflating: dataset/Train_set/with fire (4).jpg \n", " inflating: dataset/Train_set/with fire

(4).png \n", " inflating: dataset/Train_set/with fire (40).jpg \n"," inflating: dataset/Train_set/with fire (41).jpg \n"," inflating: dataset/Train_set/with fire (42).jpg \n"," inflating: dataset/Train_set/with fire (43).jpg \n"," inflating: dataset/Train_set/with fire (44).jpg \n"," inflating: dataset/Train_set/with fire (45).jpg \n"," inflating: dataset/Train_set/with fire (46).jpg \n"," inflating: dataset/Train_set/with fire (47).jpg \n"," inflating: dataset/Train_set/with fire (48).jpg \n"," inflating: dataset/Train_set/with fire (49).jpg \n"," inflating: dataset/Train_set/with fire (5).jpg \n", " inflating: dataset/Train_set/with fire (50).jpg \n"," inflating: dataset/Train_set/with fire (51).jpg \n"," inflating:

dataset/Train_set/with fire (52).jpg \n"," inflating: dataset/Train_set/with fire (53).jpg \n"," inflating: dataset/Train_set/with fire (54).jpg \n","inflating: dataset/Train_set/with fire (55).jpg \n"," inflating: dataset/Train_set/with fire (56).jpg \n"," inflating: dataset/Train_set/with fire (57).jpg \n"," inflating: dataset/Train_set/with fire (58).jpg \n","inflating: dataset/Train_set/with fire (59).jpg \n"," inflating: dataset/Train_set/with fire (6).jpg \n", " inflating: dataset/Train_set/with fire (60).jpg \n"," inflating: dataset/Train_set/with fire (61).jpg \n","inflating: dataset/Train_set/with fire (62).jpg \n"," inflating: dataset/Train_set/with fire (63).jpg \n"," inflating: dataset/Train_set/with fire (64).jpg \n"," inflating: dataset/Train_set/with fire (65).jpg \n",

" inflating: dataset/Train_set/with fire (66).jpg \n"," inflating: dataset/Train_set/with fire (67).jpg \n"," inflating: dataset/Train_set/with fire (68).jpg \n"," inflating: dataset/Train_set/with fire (69).jpg \n"," inflating: dataset/Train_set/with fire (7).jpg \n", " inflating: dataset/Train_set/with fire (70).jpg \n"," inflating: dataset/Train_set/with fire (71).jpg \n"," inflating: dataset/Train_set/with fire (72).jpg \n"," inflating: dataset/Train_set/with fire (73).jpg \n"," inflating: dataset/Train_set/with fire (74).jpg \n"," inflating: dataset/Train_set/with fire (75).jpg \n"," inflating: dataset/Train_set/with fire (76).jpg \n"," inflating: dataset/Train_set/with fire (77).jpg \n"," inflating: dataset/Train_set/with fire (78).jpg \n"," inflating: dataset/Train_set/with fire (79).jpg \n"," inflating: dataset/Train_set/with fire (8).jpg \n", " inflating: dataset/Train_set/with fire (80).jpg \n"," inflating: dataset/Train_set/with fire (81).jpg \n"," inflating: dataset/Train_set/with fire (82).jpg \n"," inflating: dataset/Train_set/with fire (83).jpg \n"," inflating: dataset/Train_set/with fire (84).jpg \n"," inflating: dataset/Train_set/with fire (85).jpg \n"," inflating: dataset/Train_set/with fire (86).jpg \n"," inflating: dataset/Train_set/with fire (87).jpg \n"," inflating: dataset/Train_set/with fire (88).jpg \n"," inflating: dataset/Train_set/with fire (89).jpg \n"," inflating: dataset/Train_set/with fire (9).jpg \n", " inflating:

dataset/Train_set/with fire (90).jpg \n","
inflating: dataset/Train_set/with fire
(91).jpg \n"," inflating:
dataset/Train_set/with fire (92).jpg \n","
inflating: dataset/Train_set/with fire
(93).jpg \n"," inflating:
dataset/Train_set/with fire (94).jpg \n","
inflating: dataset/Train_set/with fire
(95).jpg \n"," inflating:
dataset/Train_set/with fire (96).jpg \n","
inflating: dataset/Train_set/with fire
(97).jpg \n"," inflating:
dataset/Train_set/with fire (98).jpg \n","
inflating: dataset/Train_set/with fire
(99).jpg \n"," inflating:
dataset/Train_set/with fire.jpeg \n",
" inflating: dataset/Train_set/with_fire (1).gif \n",
" inflating: dataset/Train_set/with_fire
(1).jpeg \n"," inflating:
dataset/Train_set/with_fire (1).jpg \n", "
inflating: dataset/Train_set/with_fire
(1).png \n",

" inflating: dataset/Train_set/with_fire (10).jpeg
\n"," inflating: dataset/Train_set/with_fire (10).jpg
\n", " inflating: dataset/Train_set/with_fire
(100).jpg \n"," inflating:
dataset/Train_set/with_fire (101).jpg \n","
inflating: dataset/Train_set/with_fire (102).jpg
\n"," inflating: dataset/Train_set/with_fire
(103).jpg \n","inflating:
dataset/Train_set/with_fire (104).jpg \n","
inflating: dataset/Train_set/with_fire (105).jpg
\n","inflating: dataset/Train_set/with_fire
(106).jpg \n"," inflating:
dataset/Train_set/with_fire (107).jpg \n","
inflating: dataset/Train_set/with_fire (108).jpg
\n"," inflating: dataset/Train_set/with_fire
(109).jpg \n","inflating:
dataset/Train_set/with_fire (11).jpg \n", "
inflating: dataset/Train_set/with_fire (110).jpg
\n","inflating: dataset/Train_set/with_fire

(111).jpg \n"," inflating:
dataset/Train_set/with_fire (112).jpg \n","
inflating: dataset/Train_set/with_fire (113).jpg
\n"," inflating: dataset/Train_set/with_fire
(114).jpg \n","inflating:
dataset/Train_set/with_fire (115).jpg \n","
inflating: dataset/Train_set/with_fire (116).jpg
\n","inflating: dataset/Train_set/with_fire
(117).jpg \n"," inflating:
dataset/Train_set/with_fire (118).jpg \n","
inflating: dataset/Train_set/with_fire (119).jpg
\n"," inflating: dataset/Train_set/with_fire (12).jpg
\n", " inflating: dataset/Train_set/with_fire
(120).jpg \n"," inflating:
dataset/Train_set/with_fire (121).jpg \n","
inflating: dataset/Train_set/with_fire (122).jpg
\n"," inflating:
dataset/Train_set/with_fire (123).jpg \n","
inflating: dataset/Train_set/with_fire (124).jpg
\n"," inflating: dataset/Train_set/with_fire
(125).jpg \n","inflating:
dataset/Train_set/with_fire (126).jpg \n","
inflating: dataset/Train_set/with_fire (127).jpg
\n","inflating: dataset/Train_set/with_fire
(128).jpg \n"," inflating:
dataset/Train_set/with_fire (129).jpg \n","
inflating: dataset/Train_set/with_fire (13).jpg \n",
" inflating: dataset/Train_set/with_fire (130).jpg
\n","inflating: dataset/Train_set/with_fire
(131).jpg \n"," inflating:
dataset/Train_set/with_fire (132).jpg \n","
inflating: dataset/Train_set/with_fire (133).jpg
\n"," inflating: dataset/Train_set/with_fire
(134).jpg \n","inflating:
dataset/Train_set/with_fire (135).jpg \n","
inflating: dataset/Train_set/with_fire (136).jpg
\n",

" inflating: dataset/Train_set/with_fire (137).jpg \n"," inflating: dataset/Train_set/with_fire (138).jpg \n"," inflating: dataset/Train_set/with_fire (139).jpg \n"," inflating: dataset/Train_set/with_fire (14).jpg \n", " inflating: dataset/Train_set/with_fire (140).jpg \n"," inflating: dataset/Train_set/with_fire (141).jpg \n"," inflating: dataset/Train_set/with_fire (142).jpg \n"," inflating: dataset/Train_set/with_fire (143).jpg \n"," inflating: dataset/Train_set/with_fire (144).jpg \n"," inflating: dataset/Train_set/with_fire (145).jpg \n"," inflating: dataset/Train_set/with_fire (146).jpg \n",
" inflating: dataset/Train_set/with_fire (147).JPG \n", " inflating: dataset/Train_set/with_fire (148).jpg \n", " inflating: dataset/Train_set/with_fire (149).jpg \n", " inflating: dataset/Train_set/with_fire (15).jpg \n",
" inflating: dataset/Train_set/with_fire (150).jpg \n", " inflating: dataset/Train_set/with_fire (151).jpg \n",
" inflating: dataset/Train_set/with_fire (152).jpg \n", " inflating: dataset/Train_set/with_fire (153).jpg \n",
" inflating: dataset/Train_set/with_fire (154).jpg \n", " inflating: dataset/Train_set/with_fire (155).jpg \n",
" inflating: dataset/Train_set/with_fire (156).jpg \n", " inflating: dataset/Train_set/with_fire (157).jpg \n",
" inflating: dataset/Train_set/with_fire (158).jpg \n", " inflating: dataset/Train_set/with_fire (159).jpg \n",
" inflating: dataset/Train_set/with_fire (16).jpg \n",
" inflating: dataset/Train_set/with_fire

(160).JPG \n", " inflating: dataset/Train_set/with_fire (161).jpg \n", " inflating: dataset/Train_set/with_fire (162).jpg \n", " inflating: dataset/Train_set/with_fire (163).jpg \n", " inflating: dataset/Train_set/with_fire (164).jpg \n", " inflating: dataset/Train_set/with_fire (165).JPG \n", " inflating: dataset/Train_set/with_fire (166).jpg \n", " inflating: dataset/Train_set/with_fire (167).jpg \n", " inflating: dataset/Train_set/with_fire (168).jpg \n", " inflating: dataset/Train_set/with_fire (169).jpg \n", " inflating: dataset/Train_set/with_fire (17).jpg \n",
" inflating: dataset/Train_set/with_fire (170).jpg \n", " inflating: dataset/Train_set/with_fire (171).jpg \n", " inflating: dataset/Train_set/with_fire (172).jpg \n", " inflating: dataset/Train_set/with_fire (173).jpg \n", " inflating: dataset/Train_set/with_fire (174).jpg \n",

" inflating: dataset/Train_set/with_fire (175).jpg \n"," inflating: dataset/Train_set/with_fire (176).jpg \n","inflating: dataset/Train_set/with_fire (177).jpg \n"," inflating: dataset/Train_set/with_fire (178).jpg \n","inflating: dataset/Train_set/with_fire (179).jpg \n"," inflating: dataset/Train_set/with_fire (18).jpg \n", " inflating: dataset/Train_set/with_fire (180).jpg \n"," inflating: dataset/Train_set/with_fire (181).jpg \n","inflating: dataset/Train_set/with_fire (182).jpg \n"," inflating: dataset/Train_set/with_fire (183).jpg \n","inflating: dataset/Train_set/with_fire (184).jpg \n"," inflating: dataset/Train_set/with_fire (185).jpg \n"," inflating: dataset/Train_set/with_fire (186).jpg

\n"," inflating: dataset/Train_set/with_fire
(187).jpg \n","inflating:
dataset/Train_set/with_fire (188).jpg \n","
inflating: dataset/Train_set/with_fire (189).jpg
\n","inflating: dataset/Train_set/with_fire (19).jpg
\n", " inflating: dataset/Train_set/with_fire
(190).jpg \n","inflating:
dataset/Train_set/with_fire (191).jpg \n","
inflating: dataset/Train_set/with_fire (192).jpg
\n","inflating: dataset/Train_set/with_fire
(193).jpg \n"," inflating:
dataset/Train_set/with_fire (194).jpg \n","
inflating: dataset/Train_set/with_fire (195).jpg
\n"," inflating: dataset/Train_set/with_fire
(196).jpg \n","inflating:
dataset/Train_set/with_fire (197).jpg \n","
inflating: dataset/Train_set/with_fire (198).jpg
\n","inflating: dataset/Train_set/with_fire
(199).jpg \n"," inflating:
dataset/Train_set/with_fire (2).jpeg \n", "
inflating: dataset/Train_set/with_fire (2).jpg \n",
" inflating: dataset/Train_set/with_fire (2).png \n",
" inflating: dataset/Train_set/with_fire
(20).jpg \n", " inflating:
dataset/Train_set/with_fire (200).jpg \n","
inflating: dataset/Train_set/with_fire
(201).jpg \n"," inflating:
dataset/Train_set/with_fire (202).jpg \n","
inflating: dataset/Train_set/with_fire
(203).jpg \n"," inflating:
dataset/Train_set/with_fire (204).jpg \n","
inflating: dataset/Train_set/with_fire
(205).jpg \n"," inflating:
dataset/Train_set/with_fire (206).jpg \n","
inflating: dataset/Train_set/with_fire
(207).jpg \n"," inflating:
dataset/Train_set/with_fire (208).jpg \n","
inflating: dataset/Train_set/with_fire
(209).jpg \n"," inflating:
dataset/Train_set/with_fire (21).jpg \n",

" inflating: dataset/Train_set/with_fire (210).jpg \n", " inflating: dataset/Train_set/with_fire (211).jpg \n", " inflating: dataset/Train_set/with_fire (212).jpg \n", " inflating: dataset/Train_set/with_fire (213).jpg \n", " inflating: dataset/Train_set/with_fire (214).jpg \n", " inflating: dataset/Train_set/with_fire (215).jpg \n", " inflating: dataset/Train_set/with_fire (216).jpg \n", " inflating: dataset/Train_set/with_fire (217).jpg \n", " inflating: dataset/Train_set/with_fire (218).jpg \n", " inflating: dataset/Train_set/with_fire (219).jpg \n", " inflating: dataset/Train_set/with_fire (22).jpg \n",

" inflating: dataset/Train_set/with_fire (220).JPG \n", " inflating: dataset/Train_set/with_fire (221).jpg \n", " inflating: dataset/Train_set/with_fire (222).jpg \n", " inflating: dataset/Train_set/with_fire (223).jpg \n", " inflating: dataset/Train_set/with_fire (224).jpg \n", " inflating: dataset/Train_set/with_fire (225).jpg \n", " inflating: dataset/Train_set/with_fire (226).jpg \n", " inflating: dataset/Train_set/with_fire (227).jpg \n", " inflating: dataset/Train_set/with_fire (228).jpg \n", " inflating: dataset/Train_set/with_fire (229).jpg \n", " inflating: dataset/Train_set/with_fire (23).jpg \n",

" inflating: dataset/Train_set/with_fire (230).jpg \n"," inflating: dataset/Train_set/with_fire (231).jpg \n"," inflating: dataset/Train_set/with_fire (232).jpg \n"," inflating: dataset/Train_set/with_fire (233).jpg \n"," inflating: dataset/Train_set/with_fire

(234).jpg \n"," inflating:
dataset/Train_set/with_fire (235).jpg \n","
inflating: dataset/Train_set/with_fire
(236).jpg \n"," inflating:
dataset/Train_set/with_fire (237).jpg \n","
inflating: dataset/Train_set/with_fire
(238).jpg \n"," inflating:
dataset/Train_set/with_fire (239).jpg \n","
inflating: dataset/Train_set/with_fire
(24).jpg \n", " inflating:
dataset/Train_set/with_fire (240).jpg \n","
inflating: dataset/Train_set/with_fire
(241).jpg \n"," inflating:
dataset/Train_set/with_fire (242).jpg \n","
inflating: dataset/Train_set/with_fire
(243).jpg \n"," inflating:
dataset/Train_set/with_fire (244).jpg \n","
inflating: dataset/Train_set/with_fire
(245).jpg \n"," inflating:
dataset/Train_set/with_fire (246).jpg \n","
inflating: dataset/Train_set/with_fire
(247).jpg \n"," inflating:
dataset/Train_set/with_fire (248).jpg \n",

" inflating: dataset/Train_set/with_fire (249).jpg
\n"," inflating: dataset/Train_set/with_fire (25).jpg
\n", " inflating: dataset/Train_set/with_fire
(250).jpg \n"," inflating:
dataset/Train_set/with_fire (251).jpg \n","
inflating: dataset/Train_set/with_fire (252).jpg
\n"," inflating: dataset/Train_set/with_fire
(253).jpg \n","inflating:
dataset/Train_set/with_fire (254).jpg \n","
inflating: dataset/Train_set/with_fire (255).jpg
\n","inflating: dataset/Train_set/with_fire
(256).jpg \n"," inflating:
dataset/Train_set/with_fire (257).jpg \n","
inflating: dataset/Train_set/with_fire (258).jpg
\n"," inflating: dataset/Train_set/with_fire
(259).jpg \n","inflating:
dataset/Train_set/with_fire (26).jpg \n", "
inflating: dataset/Train_set/with_fire (260).jpg

\n","inflating: dataset/Train_set/with_fire
(261).jpg \n"," inflating:
dataset/Train_set/with_fire (27).jpg \n", "
inflating: dataset/Train_set/with_fire (28).jpg \n",
" inflating: dataset/Train_set/with_fire (29).jpg
\n", " inflating: dataset/Train_set/with_fire
(3).jpeg \n", " inflating: dataset/Train_set/with_fire
(3).jpg \n",
" inflating: dataset/Train_set/with_fire (3).png \n",
" inflating: dataset/Train_set/with_fire
(30).jpg \n"," inflating:
dataset/Train_set/with_fire (31).jpg \n","
inflating: dataset/Train_set/with_fire
(32).jpg \n"," inflating:
dataset/Train_set/with_fire (33).jpg \n","
inflating: dataset/Train_set/with_fire
(34).jpg \n"," inflating:
dataset/Train_set/with_fire (35).jpg \n","
inflating: dataset/Train_set/with_fire
(36).jpg \n"," inflating: dataset/Train_set/with_fire
(37).jpg \n","inflating: dataset/Train_set/with_fire
(38).jpg \n"," inflating: dataset/Train_set/with_fire
(39).jpg \n","inflating: dataset/Train_set/with_fire
(4).jpeg \n"," inflating: dataset/Train_set/with_fire
(4).jpg \n", "inflating: dataset/Train_set/with_fire
(4).png \n"," inflating: dataset/Train_set/with_fire
(40).jpg \n","inflating: dataset/Train_set/with_fire
(41).jpg \n"," inflating: dataset/Train_set/with_fire
(42).jpg \n","inflating: dataset/Train_set/with_fire
(43).jpg \n"," inflating: dataset/Train_set/with_fire
(44).jpg \n","inflating: dataset/Train_set/with_fire
(45).jpg \n"," inflating: dataset/Train_set/with_fire
(46).jpg \n","inflating: dataset/Train_set/with_fire
(47).jpg \n",

" inflating: dataset/Train_set/with_fire (48).jpg \n"," inflating: dataset/Train_set/with_fire (49).jpg \n"," inflating: dataset/Train_set/with_fire (5).jpeg \n"," inflating: dataset/Train_set/with_fire (5).jpg \n", " inflating: dataset/Train_set/with_fire (5).png \n"," inflating: dataset/Train_set/with_fire (50).jpg \n"," inflating: dataset/Train_set/with_fire (51).jpg \n"," inflating: dataset/Train_set/with_fire (52).jpg \n"," inflating: dataset/Train_set/with_fire (53).jpg \n"," inflating: dataset/Train_set/with_fire (54).jpg \n"," inflating: dataset/Train_set/with_fire (55).jpg \n"," inflating: dataset/Train_set/with_fire (56).jpg \n"," inflating: dataset/Train_set/with_fire (57).jpg \n"," inflating: dataset/Train_set/with_fire (58).jpg \n", " inflating: dataset/Train_set/with_fire (59).bmp \n"," inflating: dataset/Train_set/with_fire (59).jpg \n", " inflating: dataset/Train_set/with_fire (6).jpeg \n", " inflating: dataset/Train_set/with_fire (6).jpg \n", " inflating: dataset/Train_set/with_fire (6).png \n"," inflating: dataset/Train_set/with_fire (60).jpg \n"," inflating: dataset/Train_set/with_fire (61).jpg \n"," inflating: dataset/Train_set/with_fire (62).jpg \n"," inflating: dataset/Train_set/with_fire (63).jpg \n"," inflating: dataset/Train_set/with_fire (64).jpg \n"," inflating: dataset/Train_set/with_fire (65).jpg \n"," inflating: dataset/Train_set/with_fire (66).jpg \n"," inflating: dataset/Train_set/with_fire (67).jpg \n"," inflating:

dataset/Train_set/with_fire (68).jpg \n","
inflating: dataset/Train_set/with_fire
(69).jpg \n"," inflating:
dataset/Train_set/with_fire (7).jpeg \n","
inflating: dataset/Train_set/with_fire
(7).jpg \n", " inflating:
dataset/Train_set/with_fire (7).png \n","
inflating: dataset/Train_set/with_fire
(70).jpg \n"," inflating:
dataset/Train_set/with_fire (71).jpg \n","
inflating: dataset/Train_set/with_fire
(72).jpg \n"," inflating:
dataset/Train_set/with_fire (73).jpg \n","
inflating: dataset/Train_set/with_fire
(74).jpg \n"," inflating:
dataset/Train_set/with_fire (75).jpg \n","
inflating: dataset/Train_set/with_fire
(76).jpg \n"," inflating:
dataset/Train_set/with_fire (77).jpg \n","
inflating: dataset/Train_set/with_fire
(78).jpg \n"," inflating:
dataset/Train_set/with_fire (79).jpg \n",

" inflating: dataset/Train_set/with_fire (8).jpeg
\n"," inflating: dataset/Train_set/with_fire
(8).jpg \n", "inflating:
dataset/Train_set/with_fire (8).png \n","
inflating: dataset/Train_set/with_fire (80).jpg
\n","inflating: dataset/Train_set/with_fire
(81).jpg \n"," inflating:
dataset/Train_set/with_fire (82).jpg \n","
inflating: dataset/Train_set/with_fire (83).jpg
\n"," inflating: dataset/Train_set/with_fire
(84).jpg \n","inflating:
dataset/Train_set/with_fire (85).jpg \n","
inflating: dataset/Train_set/with_fire (86).jpg
\n","inflating: dataset/Train_set/with_fire
(87).jpg \n"," inflating:
dataset/Train_set/with_fire (88).jpg \n","
inflating: dataset/Train_set/with_fire (89).jpg
\n"," inflating: dataset/Train_set/with_fire
(9).jpeg \n","inflating:

dataset/Train_set/with_fire (9).jpg \n", "
inflating: dataset/Train_set/with_fire (9).png
\n"," inflating: dataset/Train_set/with_fire
(90).jpg \n",
" inflating: dataset/Train_set/with_fire
(91).JPG \n", " inflating:
dataset/Train_set/with_fire (92).jpg \n", "
inflating: dataset/Train_set/with_fire
(93).jpg \n", " inflating:
dataset/Train_set/with_fire (94).jpg \n", "
inflating: dataset/Train_set/with_fire
(95).jpg \n", " inflating:
dataset/Train_set/with_fire (96).jpg \n", "
inflating: dataset/Train_set/with_fire
(97).jpg \n", " inflating:
dataset/Train_set/with_fire (98).jpg \n", "
inflating: dataset/Train_set/with_fire
(99).jpg \n"
]
}
]
},
{
"cell_type":
"code",
"source": [
"train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range
=180,zoom_range
=0.2,horizontal_flip=True)\n",
"test_datagen=ImageDataGenerator(rescale=1
./255)\n"
],
"metadata": {
"id": "A1iR_1R-cKY4"
},
"execution_count": 6,

"outputs": []
},
{
"cell_type":
"markdown",
"source": [
"#Applying ImageDataGenerator functionality to
train dataset\n","\n"
],
"metadata": {
"id": "pbYsUSI9rrd9"
}
},
{
"cell_type":
"code",
"source": [
"x_train = train_datagen.flow_from_directory(\n",
" r'/content/drive/MyDrive/IBM PROJECT/dataset/Train_set',\n",
"   target_size=(435, 116),batch_size=5,color_mode='rgb',class_mode='sparse')"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "3ou6I0e9sR7o",
"outputId": "0509055a-9579-4917-9dae-5e46d116e2eb"
},
"execution_count
": 7,"outputs": [
{
"output_type":
"stream","name":
"stdout",
"text": [
"Found 436 images belonging to 1 classes.\n"
]
}
]
},
{

"cell_type":
"markdown",
"source": [
"#Applying ImageDataGenerator functionality to test dataset"
],
"metadata": {
"id": "943_QP5BtsLF"
}
},
{
"cell_type":
"code",
"source":
["x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IB
M
PROJECT/dataset/Test_set',target_size=(128,128),batch_size=32,class_mode='binar
y')\n"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "vIgeaeLht0Sn",
"outputId": "7850fd41-d0a5-45c9-a282-71b36a85781a"
},
"execution_count
": 8,"outputs": [
{
"output_type":
"stream","name":
"stdout",
"text": [
"Found 121 images belonging to 2 classes.\n"
]
}
]
},
{
"cell_type":
"code",
"source":
["print(x_train.class_indices

)"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "EGzj9MTtCWPb",
"outputId": "66a25c8c-f302-4df1-ae78-8236f25d8c50"
},
"execution_count
": 9,"outputs": [{
"output_type":
"stream","name":
"stdout",
"text":
["{'Train_set':
0}\n"
]
}
]
},
{
"cell_type": "code",
"source":
["print(x_test.class_in
dices)"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "jTQkV3tJCc_v",
"outputId": "7e8407e7-0ab0-4545-c1e1-90ec0ac7ce9a"
},
"execution_count"
: 10,"outputs": [
{
"output_type":
"stream","name":
"stdout",
"text": [
"{'Forest with fire': 0, 'Forest without fire': 1}\n"]

}
]
},
{
"cell_type":
"code",
"source": [
"from collections import
Counter as c\n",
"c(x_train .labels)"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "MdQVcIujCgwm",
"outputId": "af1da236-db50-4a1d-c17c-e12ede0a067f"
},
"execution_count"
: 11,"outputs": [
{
"output_type":
"execute_result","data":
{"text/plain":
["Counter({0:
436})"
]
},
"metadata": {},
"execution_count
": 11
}
]
}
]
}

ADVANTAGES AND DISADVANTAGES:

ADVANTAGES

- Cheap
- Best extinguishing agent for solids

DISADVANTAGES

- Rarely allowed by the fire brigade as the primary extinguishing agent for professional use (foam and powder are allowed)
- Not frost-resistant

REFERENCES:

[1] Chaoxia, C.; Shang, W.; Zhang, F. Information-Guided Flame Detection Based on Faster R-CNN, 2020 IEEE Access 2020, 8, 58923–58932.

[2] Qingjie Zhang, Jiaolong Xu, Liang Xu and Haifeng Guo, ”Deep Convolutional Neural Networks for forest fire detection,” in Proceed ings of the 2016 International Forum on Management, Education and Information Technology Application, Atlantis Press, 2016.

[3] Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications,2018