EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRE

IBM PROJECTREPORT

Submitted By

MOHAMED ASHIK S	**Reg.No. 191043801**
JAYASURYA P M	**Reg.No. 191041033**
MOHAMED IMRAN I	**Reg.No. 191043806**
KARTHIKEYAN V G	**Reg.No. 191041040**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

MAHENDRA ENGINEERING COLLEGE

(Autonomous)

Mahendhirapuri, Mallasamudram

Namakkal Dt. - 637 503

NOVEMBER 2022

CONTENT

1 . INTRODUCTION

Wildfire detection and monitoring is required to min?imize damage to forest, wilderness and agricultural areas, and hazard to people. In particular, forest surveillance and early forest-fire detection have significant relevance. These activities have been traditionally carried out by means of lookouts in the affected areas. Experienced people in watchtowers is the most extensively used method. Surveying from an aircraft has also been done in critical seasons with high risks for wild fires. However, the high cost and the subjectivity of human surveillance play an important role. Notice that in some regions, such as the countries of southern Europe, long periods of high forest-fire risks exist.

Thus, the develop?ment of sensor technologies and automatic surveillance and detection systems is an important research and de?velopment topic. It should be noted that several sensor technologies have already been applied in forest-fire detection. Thus, satellite-based techniques using meteorological satelliteshave been proposed to collect real-time information. Detection is based on the use of NOAA AVHRR images (Illera et al., 1995). These techniques are valuable in large and homogeneous regions. However, in many areas, the detection period and the resolution provided by these systems is not yet adequate for early fire detection.

1.1 Project Overview

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done usingground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities.

1.2 Purpose

The purpose of this project is to Timely information about the appearance of fire **reduce the number of areas affected by this fire and thereby minimizes the costs of fire extinguishing and the damage caused in the woods**.The very huge area of forest is destroyed by fire every year. Monitoring of the potential risk is sand **an early detection of fire can significantly shorten the reaction time and also reduce the potential damage as well as the cost of firefighting**.Wildfires are a natural part of many environments. They are nature's way of **clearing out the dead litter on forest floors**. This allows important nutrients to return to the soil, enabling a new healthy beginning for plants and animals. Fires also play an important role in the reproduction of some plants.

2 . LITERATURE SURVEY

S.NO:	TITLE OF THE PAPER	DETAILS OF THE PAPER	OBJECTIVES	METHODOLOGY USED	TAKE AWAY
1.	Early Forest Fire Detection using Drones and Artificial Intelligence.	Published on 2019	To detect forest fires early, the proper categorization of fire and fast response from the firefighting departments.	The fire detection is based on a platform that uses Unmanned Aerial Vehicles (UAVs) which constantly patrol over potentially threatened by fire areas. The UAVs utilize the benefits from Artificial Intelligence (AI). This allows to use computer vision methods for recognition and detection of smoke or fire, based on images or video input from the drone cameras.	From this journal, we use drone cameras and UAVs, because it patrols the forest always.
2.	A review on early forest fire detection system using optical remote sensing	published on 2020	To fight forest fires occurring throughout the year with an increasing intensity in the summer and autumn periods.	Detection methods that use optical sensors or RGB cameras combine features that are related to the physical properties of flame and smoke, such as color, motion, spectral, spatial, temporal, and texture characteristics.	From this journal, we use modern optical sensor networks which are known for their long range communication capabilities and extremely suitable for sensor and telemetry applications.

3.	Developing a real-time and automatic early warning system for forest fire.	Published on 2018 IEEE	To detect forest fires causing by climatic conditions and also caused by human.	The method using here is making use of stand-alone boxes which are deployed throughout the forest. Those boxes contain different sensors and a radio module to transmit data received from these sensors. Each sensor will be tested in individually and XBee modules are configured and paired using XCTU Software.	From this journal, we use Software solutions which are used for implementing microcontroller kits and to simulate and designing circuit boards.
4.	Early Fire Detection System using wireless sensor networks.	Published on 2018 IEEE	To detect fires from huge cause of forests.	The hierarchical architecture of Wireless Sensor Networks is most efficient and extensible for dense networks which simplifies the management of the forest as well as the communication and the localization of fire and sensors.	From this journal, we use cluster heads as landmark for the rest of sensor for localization in order to define their GPS coordinates according to the cluster head's coordinate.
5.	Automatic Early Forest fire Detection based Gaussian Mixture Model.	Published 2018 IEEE	To avoid the huge damage of forest caused by fires.	Based on the slow spread of smoke, firstly a time delay parameter improves Gaussian mixture model for extracting candidate smoke regions. Then, two motion features of smoke, the rate of area change and motion style are used to select smoke regions from the candidate regions.	From this journal, we use Gaussian mixture model. Because it can reconstruct background with the advantages of small storage space, adaptive learning and good noise toleration.

2.1 Existing problem

Wildfires can disrupt transportation, communications, power and gas services, and water supply. They also lead to a deterioration of the air quality, and loss of property, crops, resources, animals and peopleHigh atmospheric temperatures and dryness (low humidity) offer favorable circumstance for a fire to start. Man made causes - Fire is caused when a source of fire like naked flame, cigarette or bidi, electric spark or any source of ignition comes into contact with inflammable material.Naturally occurring wildfires are most frequently caused by lightning. There are also volcanic, meteor, and coal-seam fires, depending on the circumstances. Human caused wildfires can be accidental, intentional (arson), or from an act of negligence.

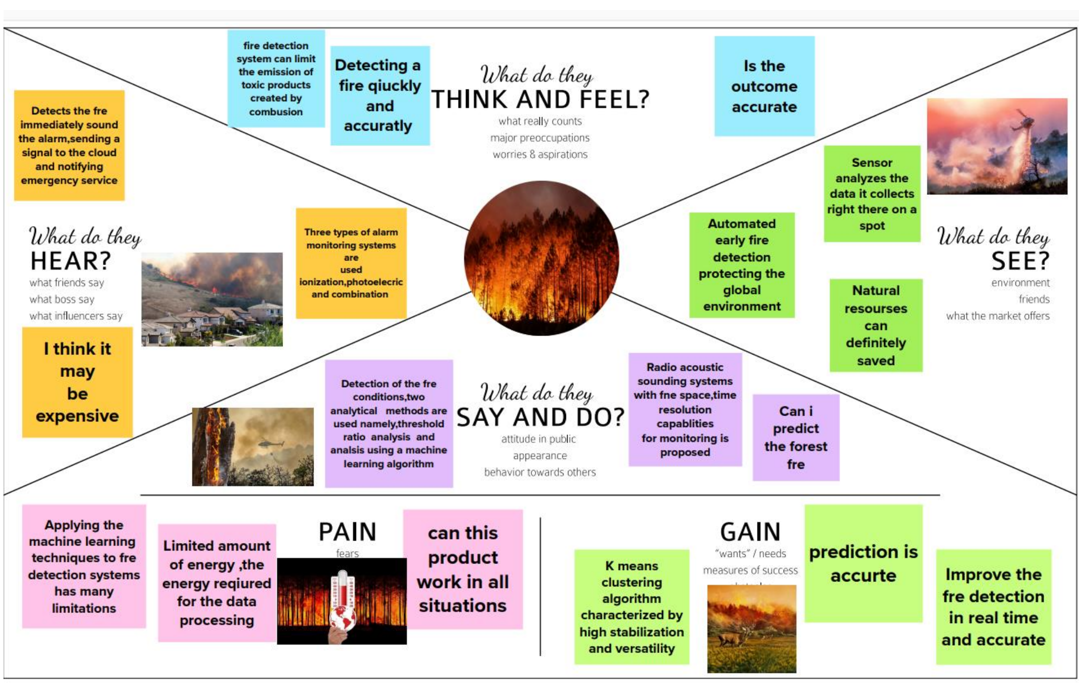
2.2 Problem Statement Definition

Problem Statement (PS)	Iam (Customer)	I’m trying to	But	Because	Which makes me feel

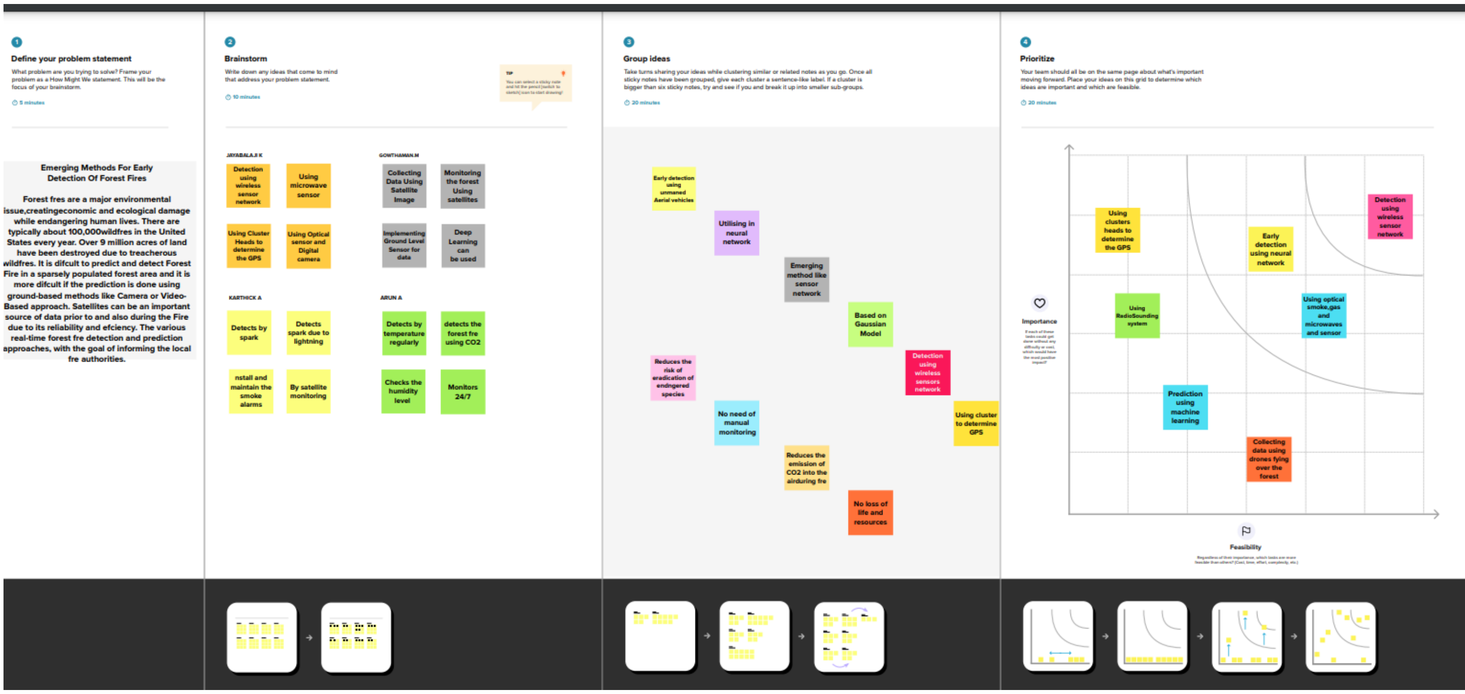
PS-1	Person near forest	Detect forest fire	Unable to know the occurrence	There is initiated in deep forest regions	Heartbroken
PS-2	Local fire authority	Know the forest fire earlier	Unable to detect earlier	There is no sensor and alarm sound fixed to sense and show warning	Worried

3 . IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



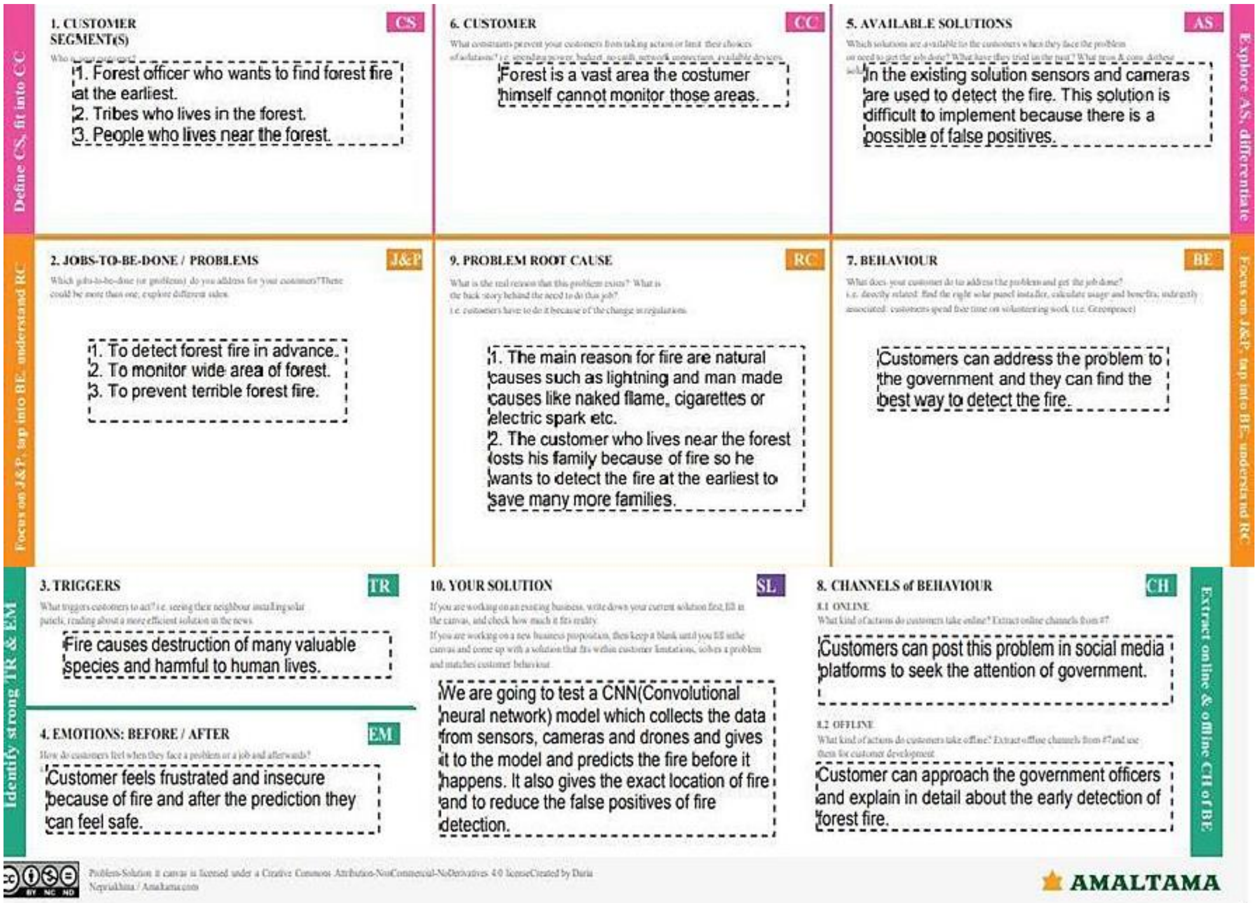
1
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
Emerging Methods For Early Detection Of Forest Fires
Forest fres are a major environmental issue,creatingeconomic and ecological damage while endangering human lives. There are typically about 100,000wildfres in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfres. It is difcult to predict and detect Forest Fire in a sparsely populated forest area and it is more difcult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efciency. The various real-time forest fre detection and prediction approaches, with the goal of informing the local fre authorities.
2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
JAYABALAJI K
Detection using wireless sensor network
Using microwave sensor
Using Cluster Heads to determine the GPS
Using Optical sensor and Digital camera
GOWTHAMAN.M
Collecting Data Using Satellite Image
Monitoring the forest Using satellites
Implementing Ground Level Sensor for data
Deep Learning can be used
KARTHICK A
Detects by spark
Detects spark due to lightning
nstall and maintain the smoke alarms
By satellite monitoring
ARUN A
Detects by temperature regulary
detects the forest fre using CO2
Checks the humidity level
Monitors 24/7
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Early detection using unmaned Aerial vehicles
Utilising in neural network
Emerging method like sensor network
Based on Gaussian Model
Detection using wireless sensors network
Using cluster to determine GPS
Reduces the risk of eradication of endngered species
No need of manual monitoring
Reduces the emission of CO2 into the airduring fre
No loss of life and resources
4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Using clusters heads to determine the GPS
Early detection using neural network
Detection using wireless sensor network
Using RadioSounding system
Using optical smoke,gas and microwaves and sensor
Prediction using machine learning
Collecting data using drones fying over the forest
Importance
Feasibility

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities
2.	Idea / Solution description	The user interacts with a web camera to read the video. Once the input image from the video frame is sent to the model, if the fire is detected it is showcased on the console, and alerting sound will be generated and an alert message will be sent to the Authorities
3.	Novelty / Uniqueness	AI assess in real time massive amounts of camera and satellite footage and identify smoke and flames from new wildfires. The systems then alert local authorities and dispatchers of the new ignition
4.	Social Impact / Customer Satisfaction	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. Tribal people who live in forest and forest department. Saving the most essential Forest cover
5.	Business Model (Revenue Model)	Supply chain, power & supply, Fires stations and government by providing services

6.	Scalability of the Solution	Tis study proposes an efective forest fre detection method using image processing techniques including movement containing region detection based on background subtraction and color segmentation. Te algorithm uses YCbCr color space which is better in separating the luminance from the chrominance and has good detection rate

3.4 Problem Solution fit



4 . REQUIREMENT ANALYSIS

4.1 Functional requirement

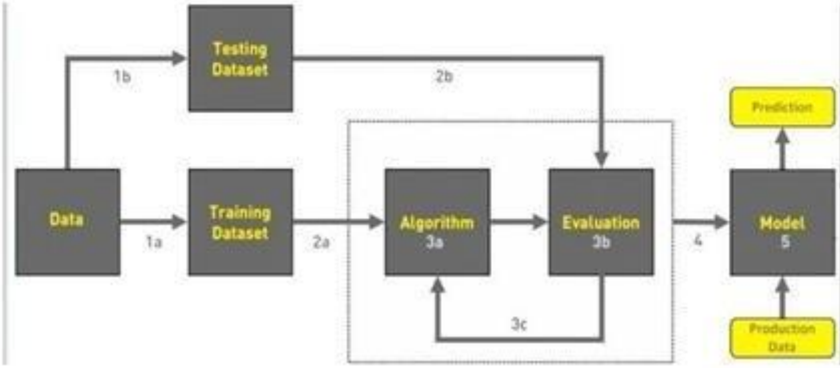
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Video surveillance start	Start surveillance through remote control
FR-2	Forest monitoring	Continuous monitoring through camera
FR-3	Detect fire	Fire is detected through CNN model
FR-4	Alert	Alert the forest officials through message

4.2 Non-Functional requirements

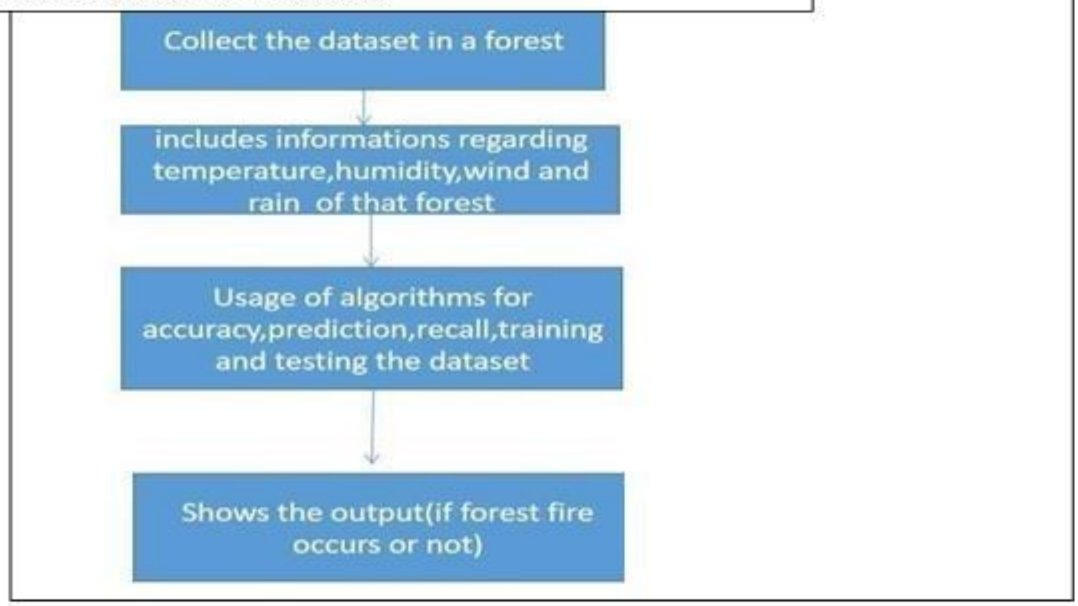
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Monitoring of the potential risk areas and an early detection of fire can significantly shorten the reaction time and also reduce the potential damage as well as the cost of fire fighting.
NFR-2	**Security**	More secure environment
NFR-3	**Reliability**	Model is safe to install
NFR-4	**Performance**	Model will achieve high accuracy
NFR-5	**Availability**	Build model is available all the time
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 is not 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

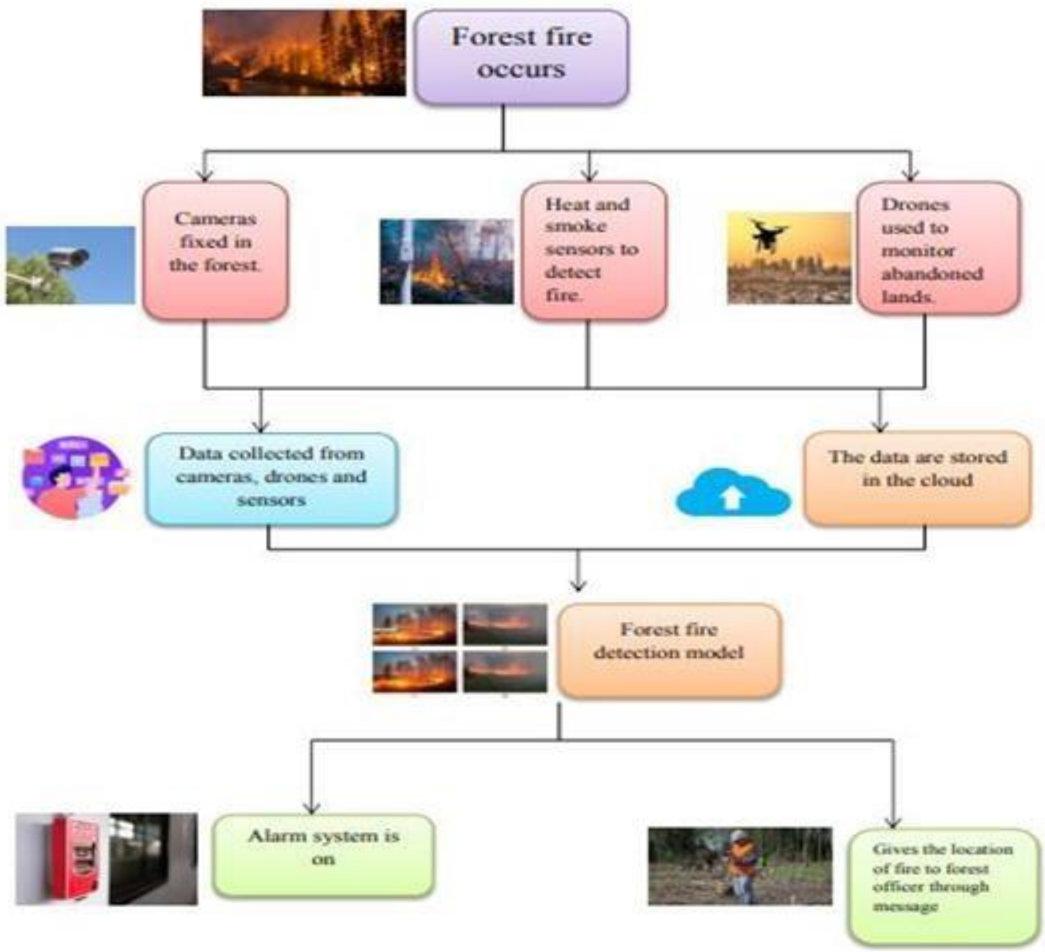
5.1 Data Flow Diagrams

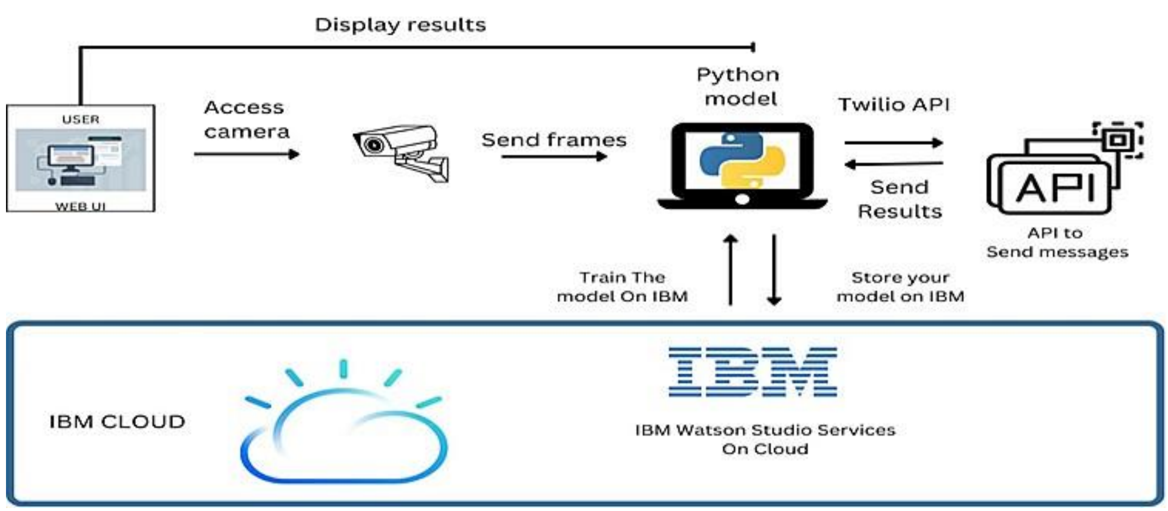


DFD Level 0 (Industry Standard)



5.2 Solution & Technical Architecture





S.No	Component	Description	Technology

1.	User Interface	The user uses the console to access the interface	Python/HTML ,CSS , Javascript andreact.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 setand 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 tosend messages	Java/python ,React.Js ,JavaScript ,HTML ,CSS ,IBMCloud ,OPEN CV ,Anaconda Navigator ,Local.

S.No	Characteristics	Description	Technology
1.	Open-SourceFrameworks	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-tierarchitecture	Web server – HTML ,CSS ,JavaScript Application server– Python , Anaconda Database server –IBM DB2
4.	Availability	Use of load balancing to distribute traffic acrossservers	IBM load balancer
5.	Performance	Enhance the performance by using IBM CDN	IBM Content Delivery Network

5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Environmentalist	Collect the data	USN-1	As an Environmentalist,it is necessary to collect the data of the forestwhich includes temperature,humidity,wind and rain of the forest	It is necessary to collectthe right data else the prediction may become wrong	High	Sprint-1
		USN-2	Identify algorithms that can be used forprediction	To collectthe algorithm to identify the accuracy levelof each algorithms	Medium	Sprint-2

	Implemen t Algorithm	USN-3	Identifythe accuracy of each algorithms	Accuracy of eachalgorithm-calculated so that it is easy to obtainthemost accurate output	High	Sprint-2
		USN-4	Evaluatethe Dataset	Data is evaluated beforeprocessin g	Medium	Sprint-1
	Evaluate Accuracyof Algorithm	USN-5	Identify accuracy,precision,recal l of eachalgorithms	These values are important for obtaining theright output	High	Sprint-3
	Display Results	USN-6	Outputs from eachalgorithm are obtained	It is highly used to predictthe effect and to take precautionary measures.	High	Sprint-4

6 . PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprin t	FunctionalRequirement (Epic)	User Story Numb er	User Story / Task	Story Points	Priorit y	Team Members

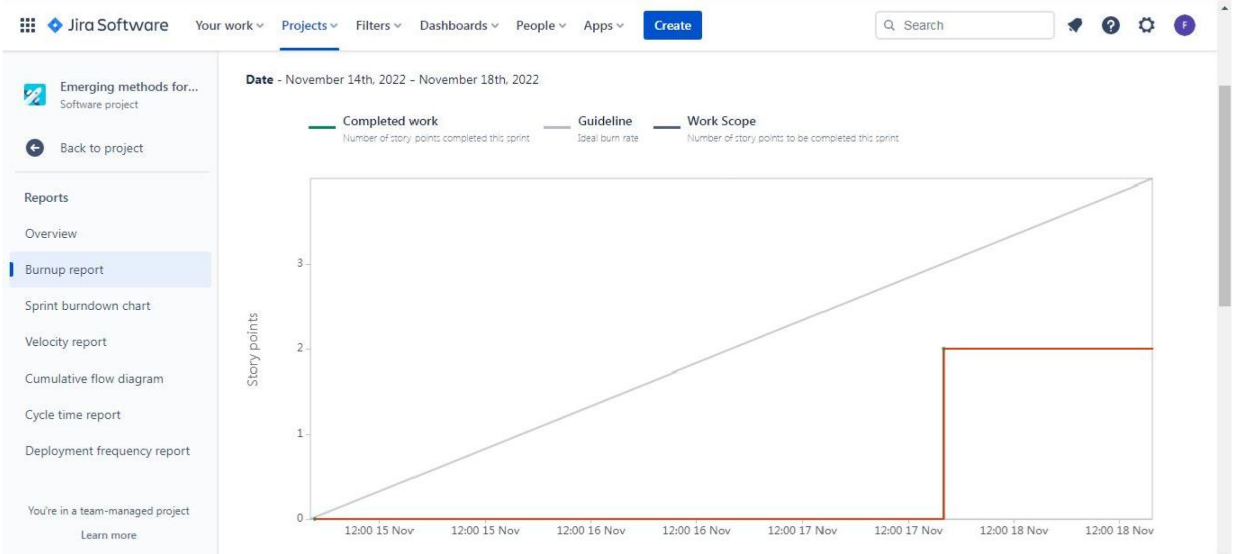
Sprint -1	Data collection	USN-1	Artificial Intelligence is a data hunger technology, it depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Convolutional Neural Networks, as it deals with images, we need training and testing data set. It is the actual data set used to train the model for performing various actions.The required datas should be collected	1	Medium	All Members

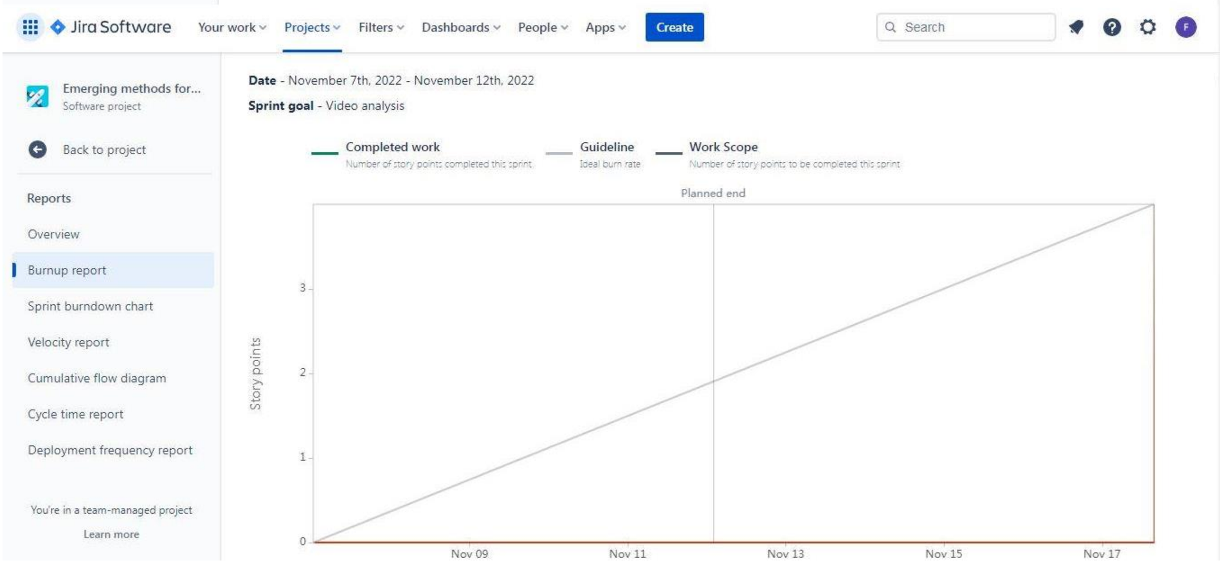
Sprint -1	Image preprocessing	USN-2	The dataset images are to be preprocessed before giving it to the model.	2	High	All Members
Sprint -2	Model building	USN-3	The dronevideos will be split intoframes todetect the fire.	3	High	All Members
Sprint-3	Video analysis	USN-4	After thefire is detected the alert message haveto be sent .	2	High	All Members
Sprint-4	Train CNN model on IBM	USN-5	The exactlocation of the drone willbe predictedandsent along withthe alert message.	2	High	All Members

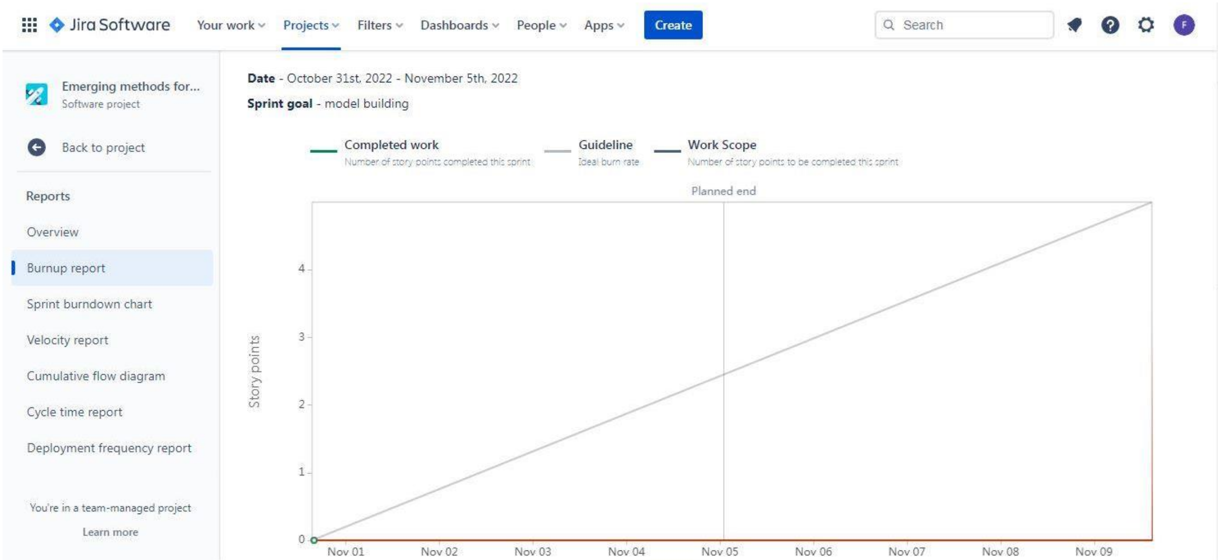
6.2 Sprint Delivery Schedule

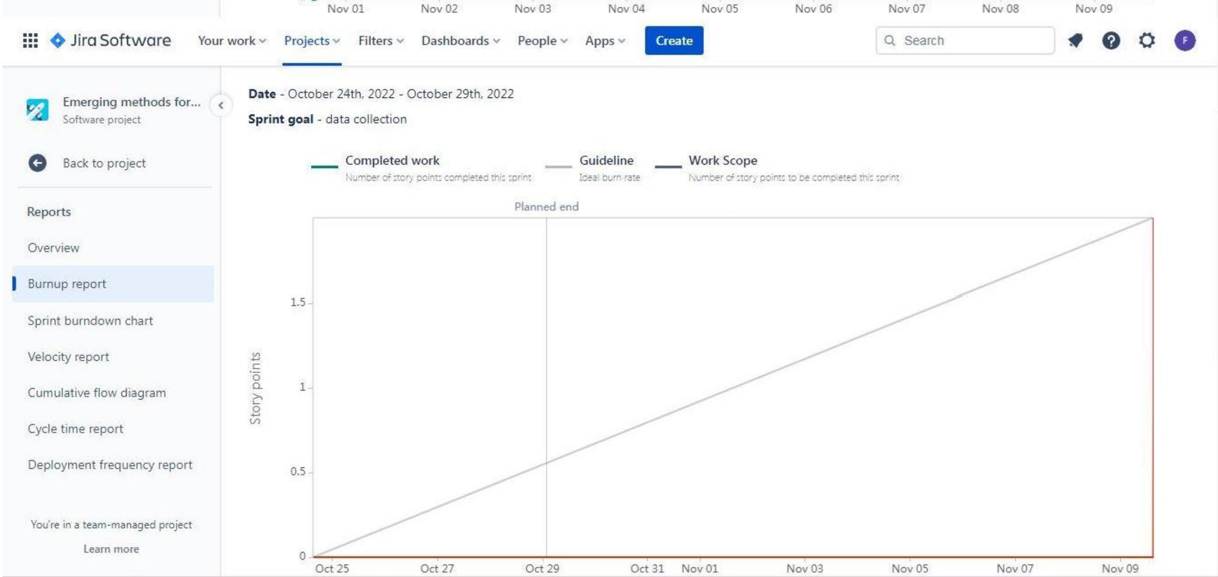
Sprint	Total Story Points	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

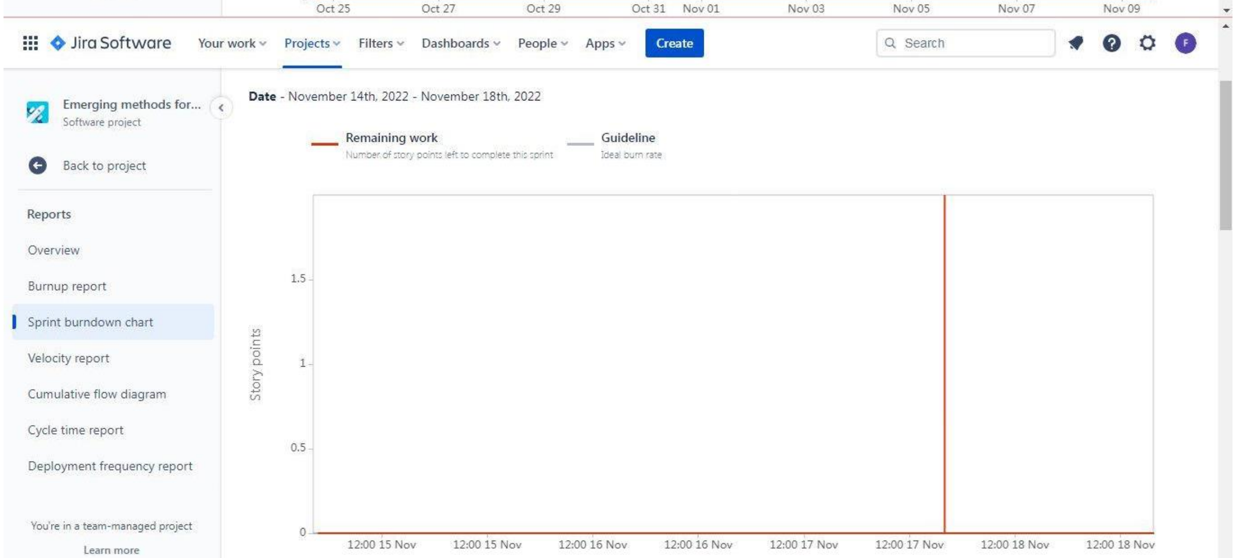
6.3 Reports from JIRA

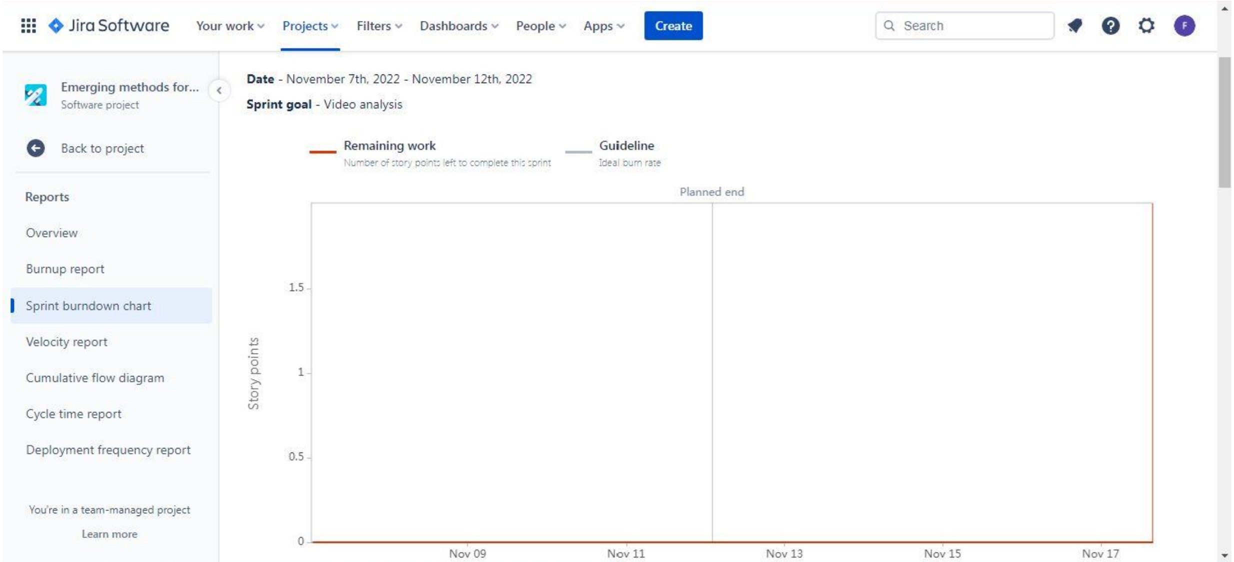


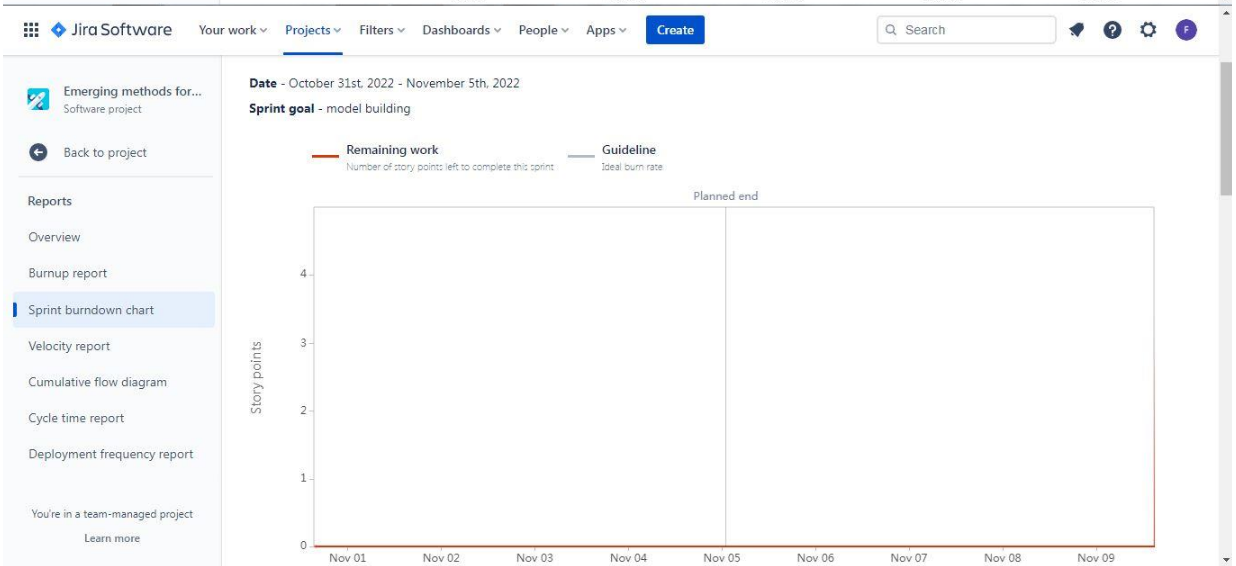


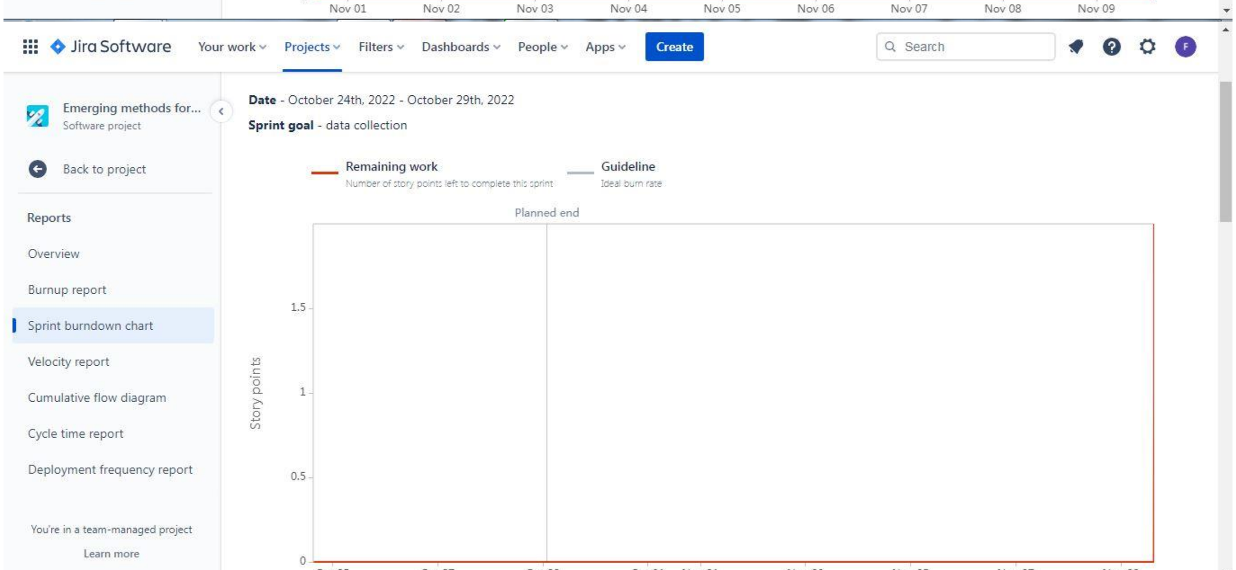

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date - October 31st, 2022 - November 5th, 2022
Sprint goal - model building
Completed work
Number of story points completed this sprint
Guideline
Ideal burn rate
Work Scope
Number of story points to be completed this sprint
Planned end
Story points
0
1
2
3
4
Nov 01
Nov 02
Nov 03
Nov 04
Nov 05
Nov 06
Nov 07
Nov 08
Nov 09

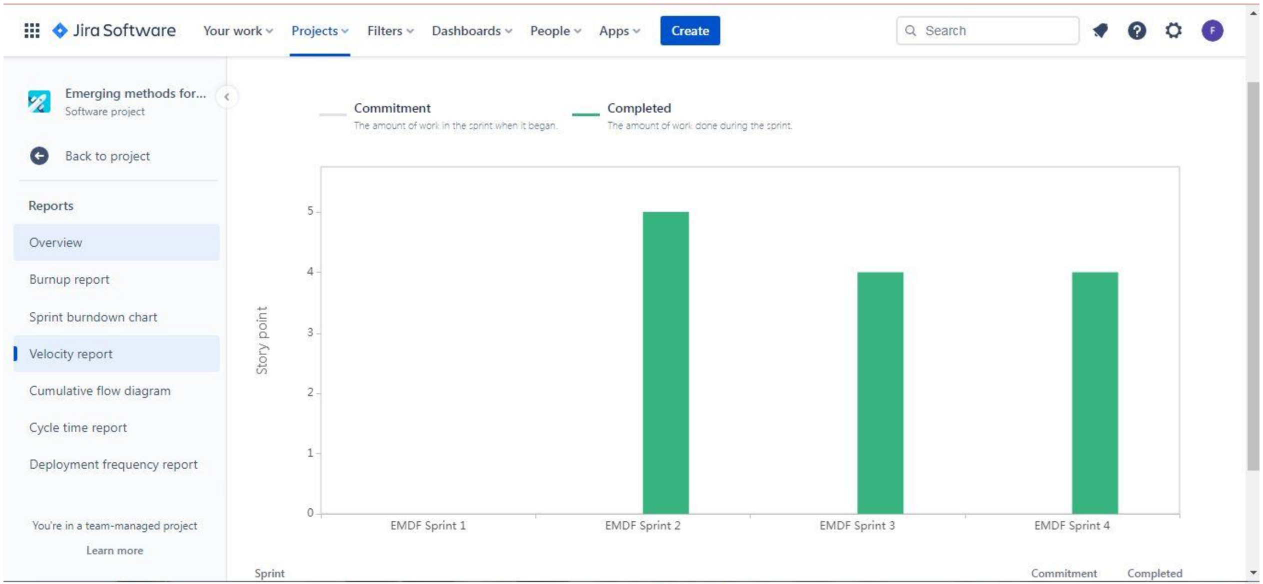

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date - October 24th, 2022 - October 29th, 2022
Sprint goal - data collection
Completed work
Number of story points completed this sprint
Guideline
Ideal burn rate
Work Scope
Number of story points to be completed this sprint
Planned end
Story points
0
0.5
1
1.5
Oct 25
Oct 27
Oct 29
Oct 31
Nov 01
Nov 03
Nov 05
Nov 07
Nov 09

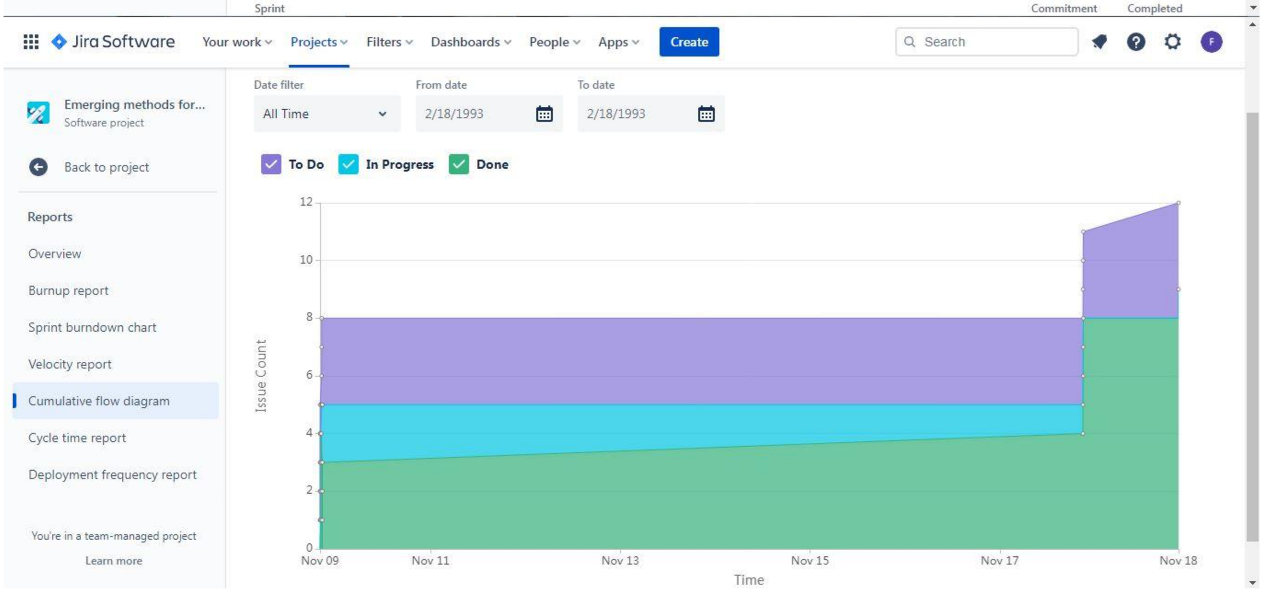

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date - November 14th, 2022 - November 18th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Story points
0
0.5
1
1.5
12:00 15 Nov
12:00 15 Nov
12:00 16 Nov
12:00 16 Nov
12:00 17 Nov
12:00 17 Nov
12:00 18 Nov
12:00 18 Nov


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date - November 7th, 2022 - November 12th, 2022
Sprint goal - Video analysis
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Planned end
Story points
1.5
1
0.5
0
Nov 09
Nov 11
Nov 13
Nov 15
Nov 17


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date - October 31st, 2022 - November 5th, 2022
Sprint goal - model building
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Planned end
Story points
4
3
2
1
0
Nov 01
Nov 02
Nov 03
Nov 04
Nov 05
Nov 06
Nov 07
Nov 08
Nov 09


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date - October 24th, 2022 - October 29th, 2022
Sprint goal - data collection
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Planned end
Story points
1.5
1
0.5
0
Oct 25
Oct 27
Oct 29
Oct 31
Nov 01
Nov 03
Nov 05
Nov 07
Nov 09


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Commitment
The amount of work in the sprint when it began.
Completed
The amount of work done during the sprint.
Story point
EMDF Sprint 1
EMDF Sprint 2
EMDF Sprint 3
EMDF Sprint 4
Sprint
Commitment
Completed


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Emerging methods for...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Date filter
All Time
From date
2/18/1993
To date
2/18/1993
To Do
In Progress
Done
Issue Count
Nov 09
Nov 11
Nov 13
Nov 15
Nov 17
Nov 18
Time

7 . CODING & SOLUTIONING

7.1 Feature 1

```
 from google.co
wlab import drive
drive.mount('/content/drive')
!pip install tensorflo
!pip install opencv-python
!pip install opencv-contrib-python
import tensorflow as tf
import numpy as np
from tensorflow import keras
import os
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)

train_dataset =
train.flow_from_directory("/content/drive/MyDrive/Dataset/Dataset/train_set",
                                          target_size=(150,150),
                                          batch_size = 32,
                                          class_mode = 'binary')

test_dataset =
test.flow_from_directory("/content/drive/MyDrive/Dataset/Dataset/test_set",
                                          target_size=(150,150),
                                          batch_size =32,
                                          class_mode = 'binary')
```

```
test_dataset.class_indices
#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
model = keras.Sequential()
model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)
))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512,activation='relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))
model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"]
)
r = model.fit(train_dataset,epochs = 10,validation_data = test_dataset)
predictions = model.predict(test_dataset)
predictions = np.round(predictions)
predictions
print(len(predictions))
model.save("forest1.h5")
```

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

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=T
```

```
[2] !pip install tensorflow
    !pip install opencv-python
    !pip install opencv-contrib-python
    import tensorflow as tf
    import numpy as np
    from tensorflow import keras
    import os
    import cv2
    from tensorflow.keras.preprocessing.image import ImageDataGenerator
    from tensorflow.keras.preprocessing import image
    import matplotlib.pyplot as plt

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.9.2)
    Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.3.0)
    Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.1.0)
    Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensc
    Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.1)
    Requirement already satisfied: tensorboard<2.10,>=2.9.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.1)
```

```
    Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (3.0.4)
[2] Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9
    Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensort
    Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->tensorflow) (3.0.9)
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (4.6.0.66)
    Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python) (1.21.6)
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: opencv-contrib-python in /usr/local/lib/python3.7/dist-packages (4.6.0.66)
    Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-contrib-python) (1.21.6)
```

```
[3] train = ImageDataGenerator(rescale=1/255)
    test = ImageDataGenerator(rescale=1/255)

    train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/Dataset/train_set",
                                              target_size=(150,150),
                                              batch_size = 32,
                                              class_mode = 'binary')

    test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/Dataset/test_set",
                                              target_size=(150,150),
                                              batch_size =32,
                                              class_mode = 'binary')

    Found 436 images belonging to 2 classes.
    Found 121 images belonging to 2 classes.
```

code 1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[4] test_dataset.class_indices
```

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

```
#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

```
model = keras.Sequential()
model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
```

0s completed at 11:52 AM

code 1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[6] model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512,activation='relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))
```

```
[7] model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```

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

```
r = model.fit(train_dataset,epochs = 10,validation_data = test_dataset)
```

```
Epoch 1/10
14/14 [==============================] - 89s 6s/step - loss: 0.6689 - accuracy: 0.6445 - val_loss: 0.6578 - val_accuracy: 0.5950
Epoch 2/10
14/14 [==============================] - 40s 3s/step - loss: 0.6210 - accuracy: 0.6445 - val_loss: 0.5821 - val_accuracy: 0.5950
Epoch 3/10
14/14 [==============================] - 40s 3s/step - loss: 0.5583 - accuracy: 0.6445 - val_loss: 0.5477 - val_accuracy: 0.5950
Epoch 4/10
14/14 [==============================] - 40s 3s/step - loss: 0.5172 - accuracy: 0.6445 - val_loss: 0.5294 - val_accuracy: 0.5950
Epoch 5/10
14/14 [==============================] - 38s 3s/step - loss: 0.5292 - accuracy: 0.6445 - val_loss: 0.5148 - val_accuracy: 0.5950
```

0s completed at 11:52 AM

code 1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[9]

```
Epoch 3/10
14/14 [==============================] - 40s 3s/step - loss: 0.5583 - accuracy: 0.6445 - val_loss: 0.5477 - val_accuracy: 0.5950
Epoch 4/10
14/14 [==============================] - 40s 3s/step - loss: 0.5172 - accuracy: 0.6445 - val_loss: 0.5294 - val_accuracy: 0.5950
Epoch 5/10
14/14 [==============================] - 38s 3s/step - loss: 0.5292 - accuracy: 0.6445 - val_loss: 0.5148 - val_accuracy: 0.5950
Epoch 6/10
14/14 [==============================] - 41s 3s/step - loss: 0.4795 - accuracy: 0.6445 - val_loss: 0.4566 - val_accuracy: 0.5950
Epoch 7/10
14/14 [==============================] - 42s 3s/step - loss: 0.4416 - accuracy: 0.6445 - val_loss: 0.4153 - val_accuracy: 0.5950
Epoch 8/10
14/14 [==============================] - 38s 3s/step - loss: 0.4363 - accuracy: 0.6445 - val_loss: 0.4070 - val_accuracy: 0.5950
Epoch 9/10
14/14 [==============================] - 39s 3s/step - loss: 0.4300 - accuracy: 0.8349 - val_loss: 0.3776 - val_accuracy: 0.9835
Epoch 10/10
14/14 [==============================] - 42s 3s/step - loss: 0.4014 - accuracy: 0.9220 - val_loss: 0.3651 - val_accuracy: 0.9917
```

[10]

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

```
4/4 [==============================] - 11s 2s/step
```

[11]

```
predictions
```

```
        [0.],
        [0.],
        [0.],
```

0s completed at 11:52 AM

code 1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

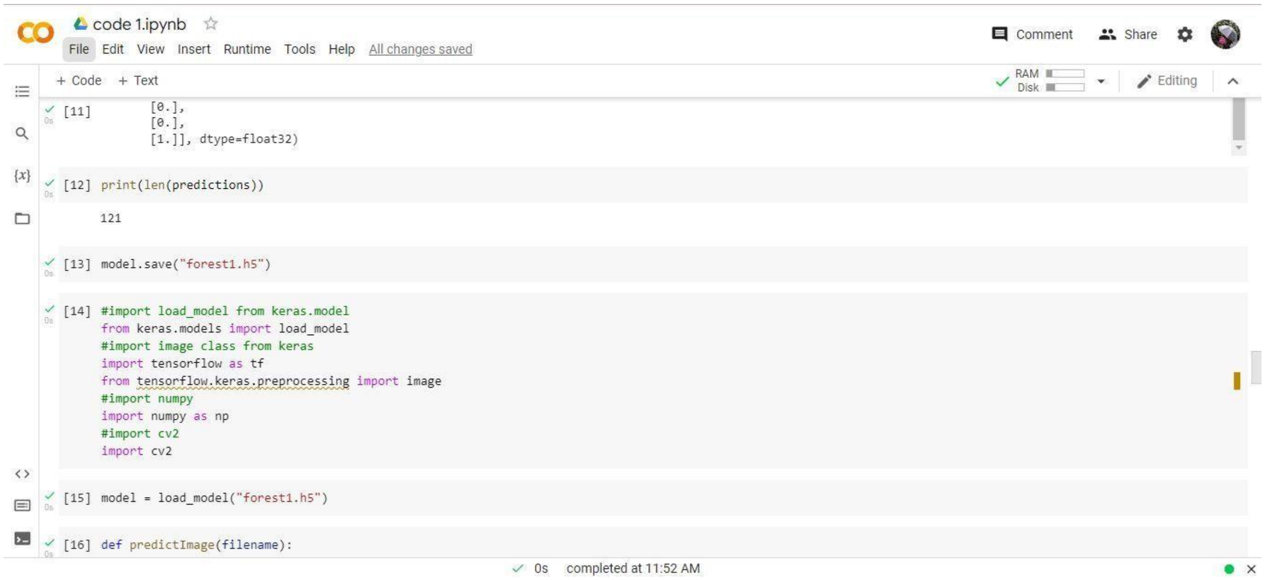
Comment Share

+ Code + Text

RAM Disk Editing

```
        [0.],
        [0.],
        [0.],
        [1.],
        [0.],
        [0.],
        [1.],
        [0.],
        [1.],
        [1.],
        [0.],
        [0.],
        [1.],
        [0.],
        [1.],
        [1.],
        [0.],
        [1.],
        [0.],
        [1.],
        [1.],
        [1.],
        [0.],
        [1.],
        [0.],
        [0.],
        [1.],
        [1.],
```

0s completed at 11:52 AM



7.2 Feature 2

```
#import load_model from keras.model
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
```

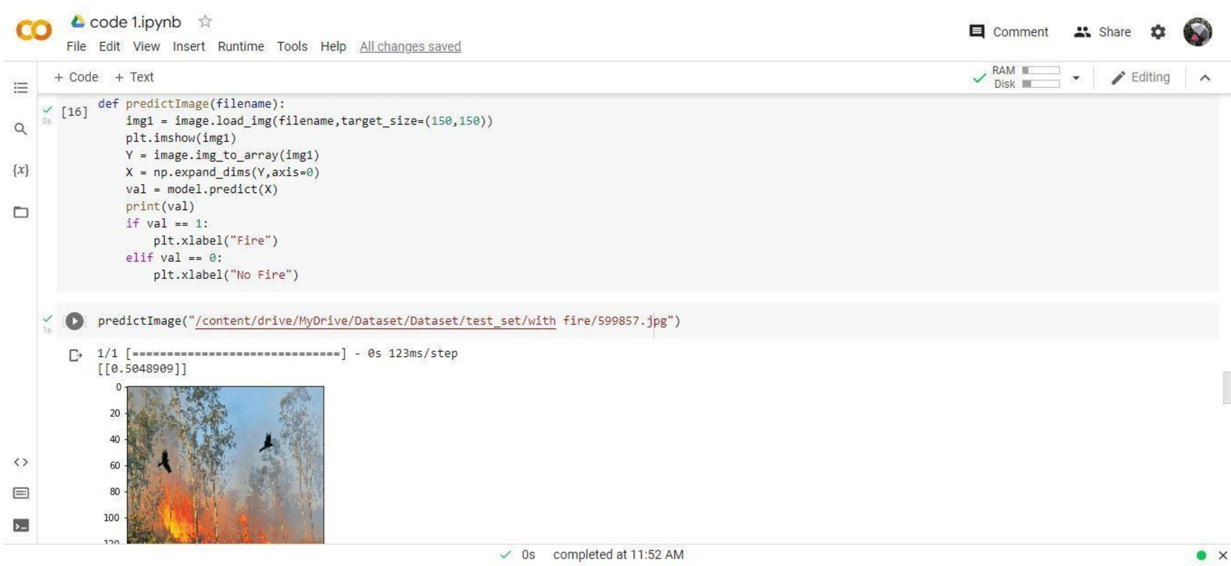
```
#import numpy
import numpy as np
#import cv2
import cv2
model = load_model("forest1.h5")
def predictImage(filename):
    img1 = image.load_img(filename,target_size=(150,150))
    plt.imshow(img1)
    Y = image.img_to_array(img1)
    X = np.expand_dims(Y,axis=0)
    val = model.predict(X)
    print(val)
    if val == 1:
        plt.xlabel("Fire")
    elif val == 0:
        plt.xlabel("No Fire")
predictImage("/content/drive/MyDrive/Dataset/Dataset/test_set/with
fire/599857.jpg")
pip install twilio
pip install playsound
#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
```

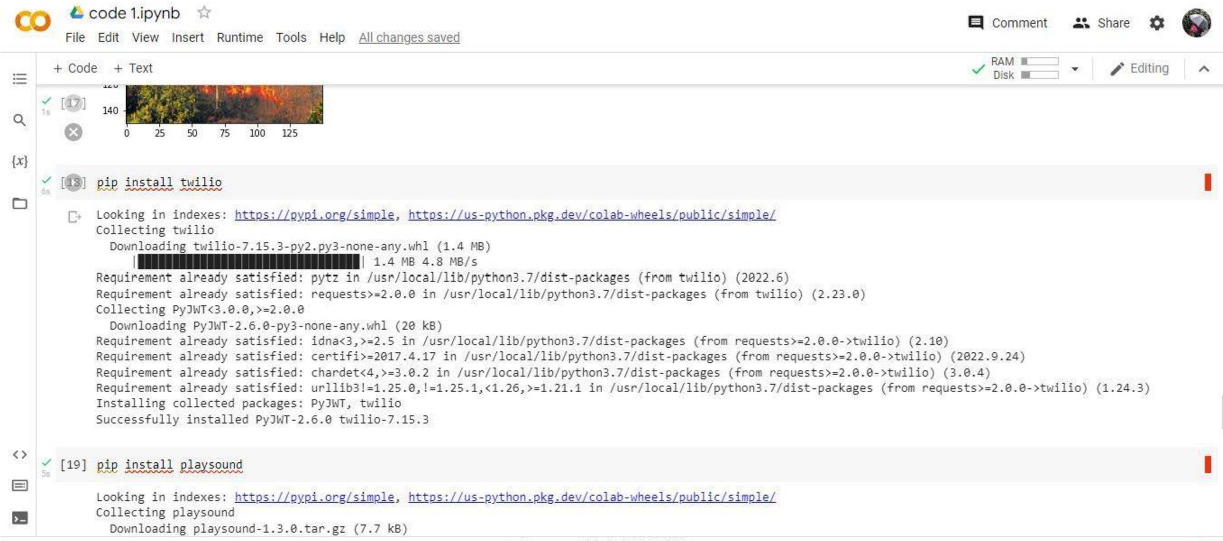
```
from twilio.rest import Client
#imort playsound package
from playsound import playsound
#load the saved model
model = load_model(r'forest1.h5')
#define video
video = cv2.VideoCapture('/content/MyDrive/Dataset/Pexels Videos
2715412.mp4')

#define the features
name = ['forest','with forest']
```

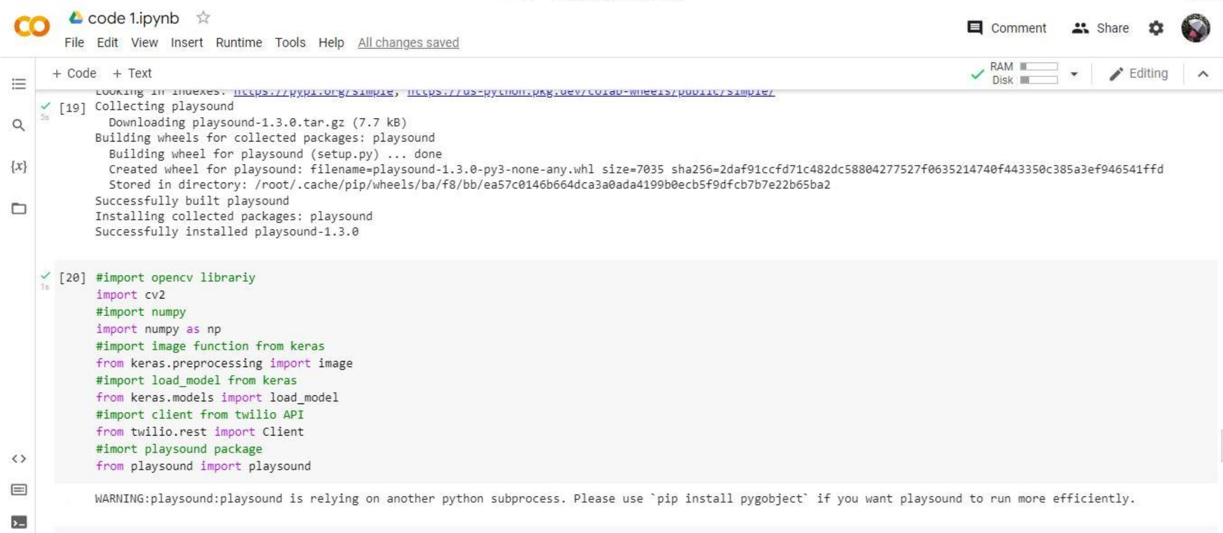
```
account_sid = 'ACd7317b90ea7039294a4e834a2731cb4a'
auth_token = '84b9f0b1f502b6ebbcbeac93b3bfad8a'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
         body='Forest fire is detected , stay alert',
         from_='+13465224352',
         to='+916383040400'

     )

print(message.sid)
```

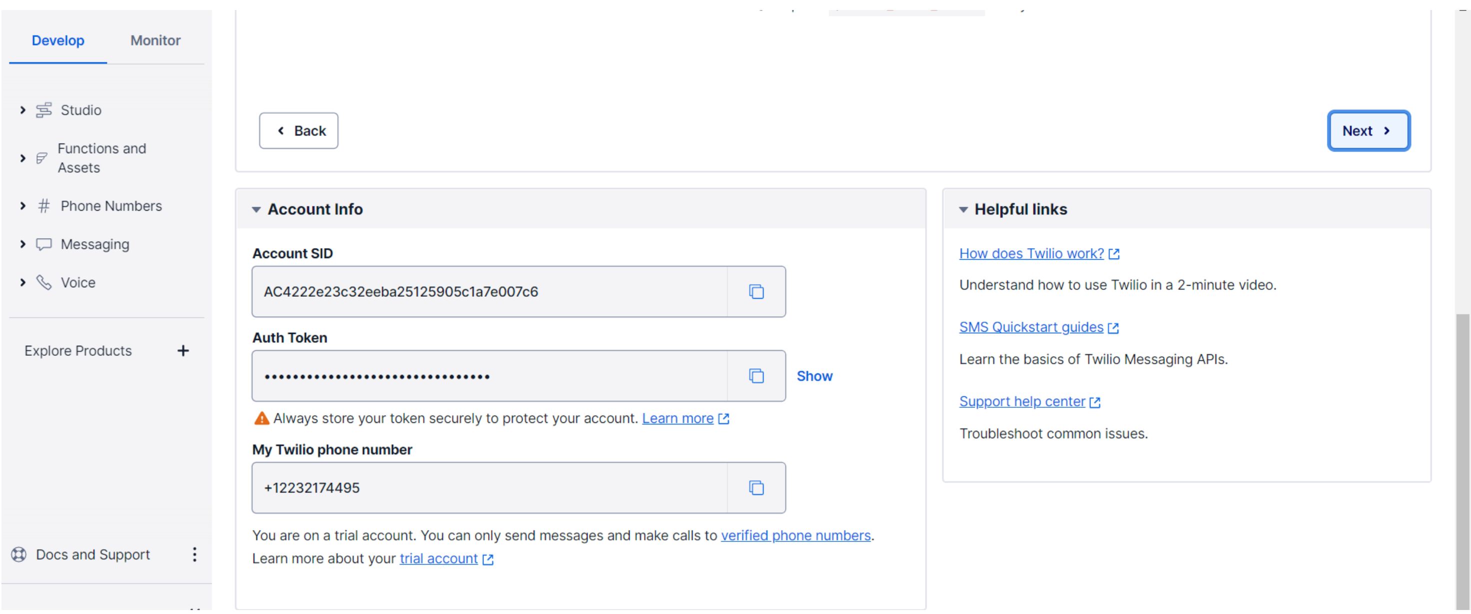
```
#import opencv library
import cv2
#import numpy
import numpy as np
#import images and load_model function from keras
from keras_preprocessing import image
```

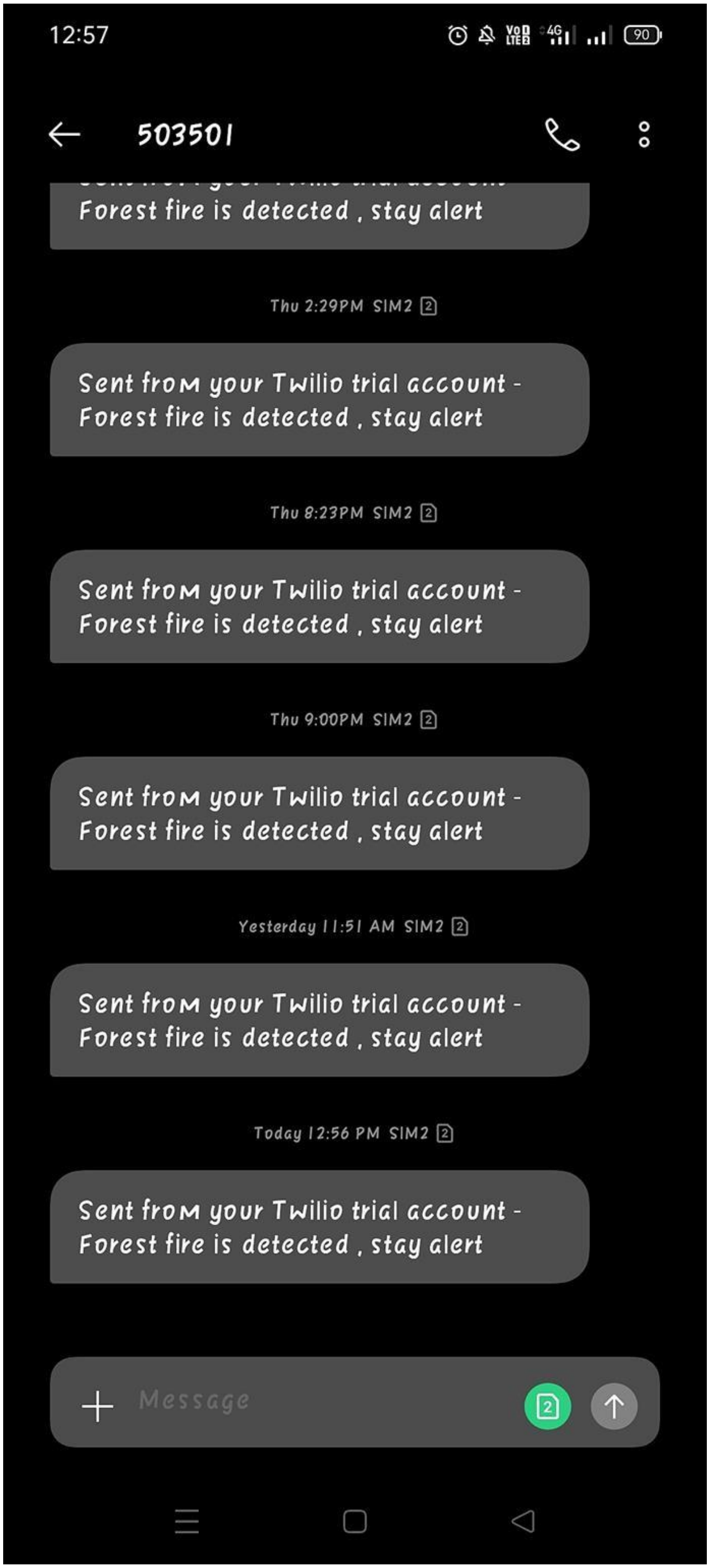
```python
from keras.models import load_model
#import client from twilio API
from  twilio.rest import Client
#import playsound package
from playsound import playsound

#load the saved model
model = load_model(r'forest1.h5')
video = cv2.VideoCapture('/content/MyDrive/Dataset/pexels-arnav-kainthola-7543653.mp4')
name = ['forest','with fire']
```




code 1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[17]
140
0 25 50 75 100 125
[18] pip install twilio
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting twilio
Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
1.4 MB 4.8 MB/s
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)
Collecting PyJWT<3.0.0,>=2.0.0
Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.24.3)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3
[19] pip install playsound
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting playsound
Downloading playsound-1.3.0.tar.gz (7.7 kB)
0s completed at 11:52 AM


code 1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[19] Collecting playsound
Downloading playsound-1.3.0.tar.gz (7.7 kB)
Building wheels for collected packages: playsound
Building wheel for playsound (setup.py) ... done
Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7035 sha256=2daf91ccfd71c482dc58804277527f0635214740f443350c385a3ef946541ffd
Stored in directory: /root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0ecb5f9dfcb7b7e22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
[20] #import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.


Develop
Monitor
Studio
Functions and Assets
Phone Numbers
Messaging
Voice
Explore Products
Docs and Support
Back
Next
Account Info
Account SID
AC4222e23c32eeba25125905c1a7e007c6
Auth Token
••••••••••••••••••••••••••••••••
Show
Always store your token securely to protect your account. Learn more
My Twilio phone number
+12232174495
You are on a trial account. You can only send messages and make calls to verified phone numbers.
Learn more about your trial account
Helpful links
How does Twilio work?
Understand how to use Twilio in a 2-minute video.
SMS Quickstart guides
Learn the basics of Twilio Messaging APIs.
Support help center
Troubleshoot common issues.


12:57
503501
Forest fire is detected , stay alert
Thu 2:29PM SIM2
Sent from your Twilio trial account - Forest fire is detected , stay alert
Thu 8:23PM SIM2
Sent from your Twilio trial account - Forest fire is detected , stay alert
Thu 9:00PM SIM2
Sent from your Twilio trial account - Forest fire is detected , stay alert
Yesterday 11:51 AM SIM2
Sent from your Twilio trial account - Forest fire is detected , stay alert
Today 12:56 PM SIM2
Sent from your Twilio trial account - Forest fire is detected , stay alert
Message

8. TESTING

8.1 Test Cases

				19.11.22						
				PNT2022TMID48153						
				Emerging methods for Early detection of forest fires						
				4 marks						
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**BUG ID**	**Executed By**
TC1	Functional	Test Page	Data collection	. It is the actual data set used to train the model for performing various actions.The required datas should be collected	data collected and kept in drive and accessed	Working as expected	Pass	data is collected successfuly		M.Annapoorani P.Dayana B.Fahmidha D.Santhi
TC2	UI	Test Page	Image preprocessing	The dataset images are to be preprocessed before giving it to the model.	Image preprocessed from tenson flow and keras	Working as expected	Pass	Image is preprocessed successfully		M.Annapoorani P.Dayana B.Fahmidha D.Santhi
TC3	Functional	Test Page	Model building	The drone videos will be split into frames todetect the fire.	Model build through keras layers	Working as expected	Pass	Model building done successfully		M.Annapoorani P.Dayana B.Fahmidha D.Santhi
TC4	Functional	Test Page	Video analysis	After the fire is detected the alert message haveto be sent .	video is analysed and alert msg is sent to given number in twilio	Working as expected	Pass	video analysis done successfully		M.Annapoorani P.Dayana B.Fahmidha D.Santhi
TC5	Functional	Test Page	Train CNN model on IBM	The exact location of the drone will be predictedand sent along with the alert message.	our code jupiter notebook is connected to IBM cloud .	Working as expected	Pass	CNN model on IBM trained successfully		M.Annapoorani P.Dayana B.Fahmidha D.Santhi

8.2 User Acceptance Testing

DefectAnalysis

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	1	4	0	9	14
Duplicate	1	1	3	5	10
External	2	2	0	0	4
Fixed	1	2	4	2	9
Not Reproduced	0	2	1	0	3

Skipped	0	0	3	1	4
Won'tFix	0	4	0	1	5
Totals	5	15	11	18	49

TestCaseAnalysis

Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	7	0	1	6
ClientApplication	51	0	2	49
Security	2	0	1	1
OutsourceShipping	3	0	1	2
ExceptionReporting	9	0	4	5
FinalReportOutput	4	0	2	2
VersionControl	2	0	0	2

9. RESULTS

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot

1.	Model Summary	**3,453,213**	model = keras.Sequential() model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3))) model.add(keras.layers.MaxPool2D(2,2)) model.add(keras.layers.Conv2D(64,(3,3),activation='relu')) model.add(keras.layers.MaxPool2D(2,2)) model.add(keras.layers.Conv2D(128,(3,3),activation='relu')) model.add(keras.layers.MaxPool2D(2,2)) model.add(keras.layers.Conv2D(128,(3,3),activation='relu')) model.add(keras.layers.MaxPool2D(2,2)) model.add(keras.layers.Flatten()) model.add(keras.layers.Dense(512,activation='relu')) model.add(keras.layers.Dense(1,activation='sigmoid')) model.add(Dense(150,activation='relu')) model.add(Dense(1,activation='sigmoid'))
2.	Accuracy	Training Accuracy – 0.9835 Validation Accuracy - 0.9917	Epoch 1/10 14/14 [==============================] - 89s 6s/step - loss: 0.6689 - accuracy: 0.6445 - val_loss: 0.6578 - val_accuracy: 0.5950 Epoch 2/10 14/14 [==============================] - 40s 3s/step - loss: 0.6210 - accuracy: 0.6445 - val_loss: 0.5821 - val_accuracy: 0.5950 Epoch 3/10 14/14 [==============================] - 40s 3s/step - loss: 0.5583 - accuracy: 0.6445 - val_loss: 0.5477 - val_accuracy: 0.5950 Epoch 4/10 14/14 [==============================] - 40s 3s/step - loss: 0.5172 - accuracy: 0.6445 - val_loss: 0.5294 - val_accuracy: 0.5950 Epoch 5/10 14/14 [==============================] - 38s 3s/step - loss: 0.5292 - accuracy: 0.6445 - val_loss: 0.5148 - val_accuracy: 0.5950 Epoch 6/10 14/14 [==============================] - 41s 3s/step - loss: 0.4795 - accuracy: 0.6445 - val_loss: 0.4566 - val_accuracy: 0.5950 Epoch 7/10 14/14 [==============================] - 42s 3s/step - loss: 0.4416 - accuracy: 0.6445 - val_loss: 0.4153 - val_accuracy: 0.5950 Epoch 8/10 14/14 [==============================] - 38s 3s/step - loss: 0.4363 - accuracy: 0.6445 - val_loss: 0.4070 - val_accuracy: 0.5950 Epoch 9/10 14/14 [==============================] - 39s 3s/step - loss: 0.4300 - accuracy: 0.8349 - val_loss: 0.3776 - val_accuracy: 0.9835 Epoch 10/10 14/14 [==============================] - 42s 3s/step - loss: 0.4014 - accuracy: 0.9220 - val_loss: 0.3651 - val_accuracy: 0.9917

8 . ADVANTAGES & DISADVANTAGES

ADVANTAGES

Timely information about the appearance of fire reduce the number of areas affected by this fire and thereby **minimizes the costs of fire extinguishing and the damage caused in the woods**.

DISADVANTAGES

Several forest fire detection methods have been implemented, such as watchtowers, satellite image processing methods, optical sensors, and digital camera-based methods[2], although there are many drawbacks, such as **inefficiency, power consumption, latency, accuracy**

11. CONCLUSION

A real-time and reliable fire detection method for an early warning system is required so that an immediate response to an incident can be made effective. In this study, methods based on color probabilities and motion features were successfully implemented to achieve this goal. The proposed method exploits the characteristics of the color of fire by developing a probability model using a multiple Gaussian. On the other hand, other fire characteristics, namely, dynamic fire movement modeled with motion features based on moment invariants, were also applied. The experiment found that the processing time required on average reached 21.70 FPS with a relatively high true positive rate of 89.92%. These results indicate that the proposed method is suitable for a real-time early warning system. Nonetheless, one of the greatest challenges in implementing the module is physically installing the camera, which may be very difficult. Therefore, it will remain a challenge for our further research.This system effectively detects and verifies the presence of fire in forest regions.The addition of Region proposals in CNN layers can result in better accuracy as well as faster execution. Our system can verify the presenceof fire in the forest with an accuracy of 97.29% from the RCNN model. This will help in the beginningphases of fire identification and assist with restricting the fire to restricted regions toprevent large-scale damage. This system focuses on observing the forests without steady human supervision.

12. FUTURE SCOPE

Detection of forest fire and smoke in wildland areas is done through remote sensing-based methods such as **satellites, high-resolution static cameras fixed on the ground, and unmanned aerial vehicles (UAVs)**.From the various inferences, it has been understood that most of the researchers have worked to increase the accuracy, and the area coverage was until 1,500 meters to the maximum. To provide more accuracy and precision, the 3D modeling of data is required, and further visualizationof forest fire images could be made very easy for interpretation. The objective of this work is to deploy an efficient and robust detection fire in the early stage. Hence, a deep learning model is required so that the boundary region could be extended, and the 3D modeling images must be considered for the prediction process to augment the accuracy.

9 . APPENDIX

Source Code

```
from google.co
wlab import drive
drive.mount('/content/drive')
!pip install tensorflo
!pip install opencv-python
!pip install opencv-contrib-python
import tensorflow as tf
import numpy as np
from tensorflow import keras
import os
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)
```

```
train_dataset =
train.flow_from_directory("/content/drive/MyDrive/Dataset/Dataset/train_set",
                                          target_size=(150,150),
                                          batch_size = 32,
                                          class_mode = 'binary')

test_dataset =
test.flow_from_directory("/content/drive/MyDrive/Dataset/Dataset/test_set",
                                          target_size=(150,150),
                                          batch_size =32,
                                          class_mode = 'binary')
test_dataset.class_indices
#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
model = keras.Sequential()
model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)
))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))
```

```
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512,activation='relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))
model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"]
)
r = model.fit(train_dataset,epochs = 10,validation_data = test_dataset)
predictions = model.predict(test_dataset)
predictions = np.round(predictions)
predictions
print(len(predictions))
model.save("forest1.h5")
#import load_model from keras.model
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2
model = load_model("forest1.h5")
def predictImage(filename):
```

```
    img1 = image.load_img(filename,target_size=(150,150))
    plt.imshow(img1)
    Y = image.img_to_array(img1)
    X = np.expand_dims(Y,axis=0)
    val = model.predict(X)
    print(val)
    if val == 1:
        plt.xlabel("Fire")
    elif val == 0:
        plt.xlabel("No Fire")
predictImage("/content/drive/MyDrive/Dataset/Dataset/test_set/with
fire/599857.jpg")
pip install twilio
pip install playsound
#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
#load the saved model
model = load_model(r'forest1.h5')
#define video
```

```
video = cv2.VideoCapture('/content/MyDrive/Dataset/Pexels Videos
2715412.mp4')

#define the features
name = ['forest','with forest']
account_sid = 'ACd7317b90ea7039294a4e834a2731cb4a'
auth_token = '84b9f0b1f502b6ebbcbeac93b3bfad8a'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
         body='Forest fire is detected , stay alert',
         from_='+13465224352',
         to='+916383040400'

     )

print(message.sid)
     )

print(message.sid)
#import opencv library
import cv2
#import numpy
import numpy as np
#import images and load_model function from keras
from keras_preprocessing import image
from keras.models import load_model
#import client from twilio API
from  twilio.rest import Client
#import playsound package
from playsound import playsound
```

```
#load the saved model
model = load_model(r'forest1.h5')
video = cv2.VideoCapture('/content/MyDrive/Dataset/pexels-arnav-kainthola-
7543653.mp4')
name = ['forest','with fire']
```

GitHub & Project Demo Link

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-7108-1658847568

References

1) Carrega, P. (1990). Climatology and index of forest fire hazard in medi?terranean France.Proc., Int. Conf. on Forest Fire Research. Coimbra. Gandia, A., Criado, A., & Rallo, M. (1994).

2) El sistema BOSQUE, alta tecnologia en defensa del medio ambiente. DYNA, no.

6, pp. 34—38 Hudson, R. D. (1969).

3) Infrared system engineering. Wiley, New York. Illera, P., Ferna´ ndez, A., & Casanova, J. L. (1995).

4) Automatic algorithm for the detection and analysis of fires by means of NOAA AVHRR images. Earsel advances in remote sensing. (Vol 4 No. 3-XII). Laurenti, A., & Neri, A. (1996).

5) Remote sensing, communications and information technologies for vegetation fire emergencies. Proc. ¹IEMEC'96, Montreal. Lim Jae, S. (1990).

6) ¹wo-dimensional signal and image processing. Prentice Hall Signal Processing Series, Prentice-Hall, Englewood Cliffs, NJ. Murillo, J. J., Ollero, A., Arru´ e, B.C., & Martı´ nez, J. R. (1997).

7) A hybrid infrared/visual system for improving reliability of fire detection systems. IFAC Symp. on Fault Detection, Supervision and Safety for ¹echnical Processes: SAFEPROCESS'97. Kingston Upon Hull, UK. Thornthwaite, C.W., & Mather, J.R. (1955). The water balance. Clima?tology, 8(1), 104. Vries, J.S., & Kemp, R.A.W. (1994). Results with a multi-spectral auton?omous wildfire detection system.

8) Proc. 2nd Conf. Forest Fire Re?search (Vol. II, pp. 779—791) C.P. 27, Coimbra,November 1994.

9)Wybo, J.L., & Jaber, A. (1996). DEDICS: a distributed environmental disaster information and control system. Proc. TIEMEC+96, Montreal