# A PROJECT REPORT ON

# EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

***Domain : ARTIFICIAL INTELLIGENCE***

***Team ID: PNT2022TMID01103***

***College Name: Panimalar Engineering College***


***R.PRANATI***
***(211419106194)***
***Department of Electronics and Communication Engineering***

***P.MONISHA***
***(211419106171)***
***Department of Electronics and Communication Engineering***

***S.PREETHI***
***(211419106204)***
***Department of Electronics and Communication Engineering***

***R.SANGEETHA PRIYA***
***(211419106233)***
***Department of Electronics and Communication Engineering***

## *TABLE OF CONTENTS*

# *1.INTRODUCTION*

## *1.1 Project Overview*

***Forests, which are diverse centers of flora and wildlife and create 1/3 of the world's oxygen, are at risk of forest fires, both natural and man-made. The precaution of averting such a massive devastating flare can save many animals and the environment. Protecting forests before they are harmed is a method of repaying Mother Nature's everlasting gift.***

***Wildfires are one of the biggest catastrophes faced by our society today causing irrevocable damages.These forest fires can be man-made or caused by mothernature by different weather conditions, torrential winds. These fires cause damages not only to the environment they also destroy vast homes and property.***

## *1.2 Purpose*

***Forest fires have become a major threat around the world, causing many negative impacts on human habitats and forest ecosystems. Climatic changes and the greenhouse effect are some of the consequences of such destruction. A higher percentage of forestfires occur due to human activities.The goal of the project is to develop a forest fire detection system that can identify forest fires in their early phases.***

# *2. LITERATURE SURVEY*

## *2.1 Existing Problem*

***Every year, there are an estimated340,000 premature deaths from respiratory and cardiovascular issues attributed to wildfire smoke.***

***The increasing frequency and severity of wildfires pose a growingthreat to biodiversity globally. Individuals, companies and public authorities bear great economic costs due to fires. In order to reduce all these, we need to detect the forest fire at an early stage and prevent it.***

## *2.2 References*


- ***Turgay Celik, Huseyin Ozkaramanl, and Hassan Demirel (2007). Fire and Smoke detection without Sensors: Image Processing based approach.15th European signal processing conference (eusipco 2007), Poznan, Poland, September 3-7.***
- ***Osman Gunay, A. Enis C, Etin, Yusuf Hakan, Habiboglu. Flame Detection method in video using Covariance descriptors, IEEE transactions, 1817-1820.***
- ***CHENG Caixia, SUN Fuchun, ZHOU Xinquan (2011).***

*One Fire Detection Method Using Neural Networks, Tsinghua Science and Technology, ISSN 1007-0214 05/17 31-35Volume 16, Number 1.*

- *S. A. Christopher, M. Wang, T. A. Berendes, and R. M. Welch (1998). The 1985 biomass burning season in South America: Satellite remote sensing of fires, smoke, and regional radiative energy budgets, vol. 37, 661– 678*
- *Paulo Vinicius Koerich Borges (2010). A Probabilistic Approach for VisionBased Fire Detection in Videos, IEEE transactions on circuits and systems for video technology, vol. 20, no. 5.*
- *Jiawei Han, Micheline Kamber, Jian Pei (2012). Data Mining Concepts and Techniques, Third edition, 248-253, 350-351.*

## *2.3 Problem Statement Definition*

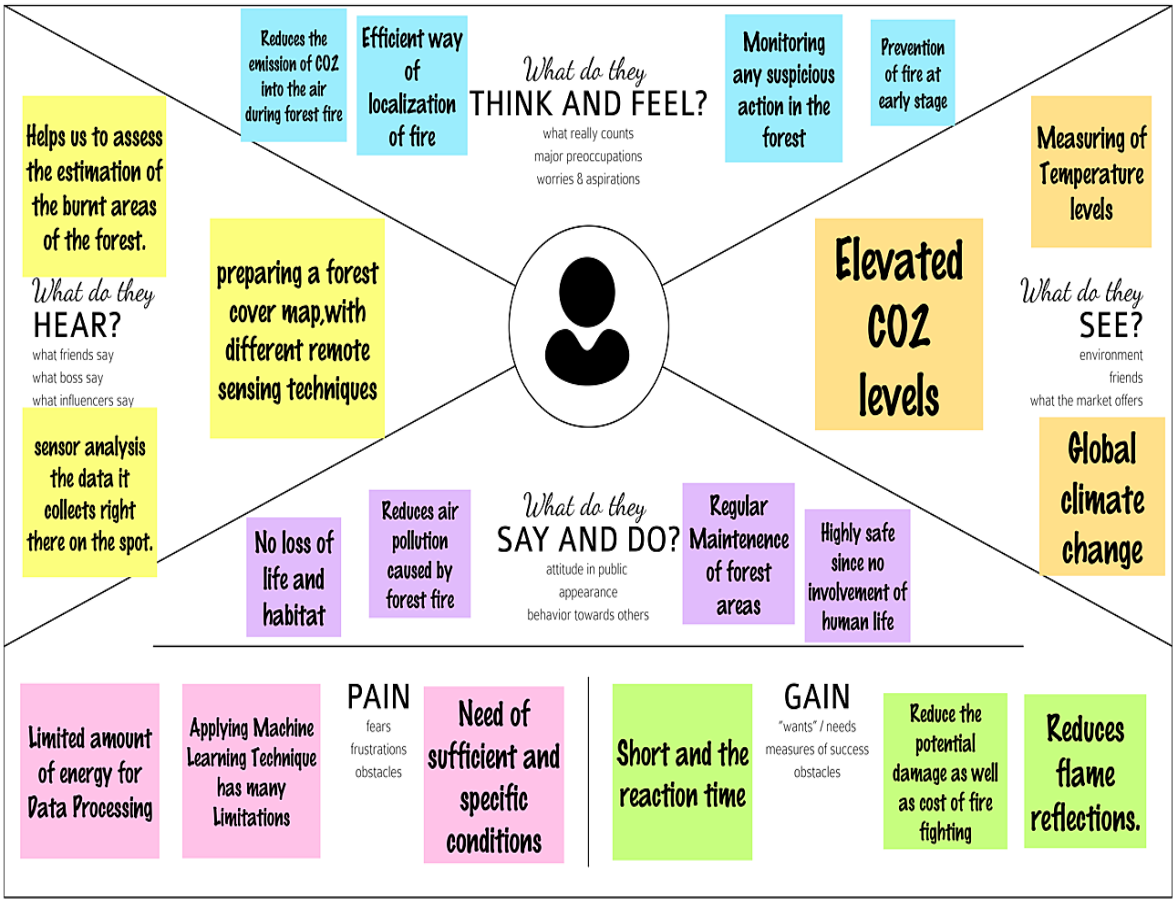
- *In the past, fires were detectedby watching towers or using satellite images.*
- *Satellites collect images of fires and send them to a monitoring authority for review. If the images*

*appear to show a fire, the authority will determine whetherthe fire isburning or not.*

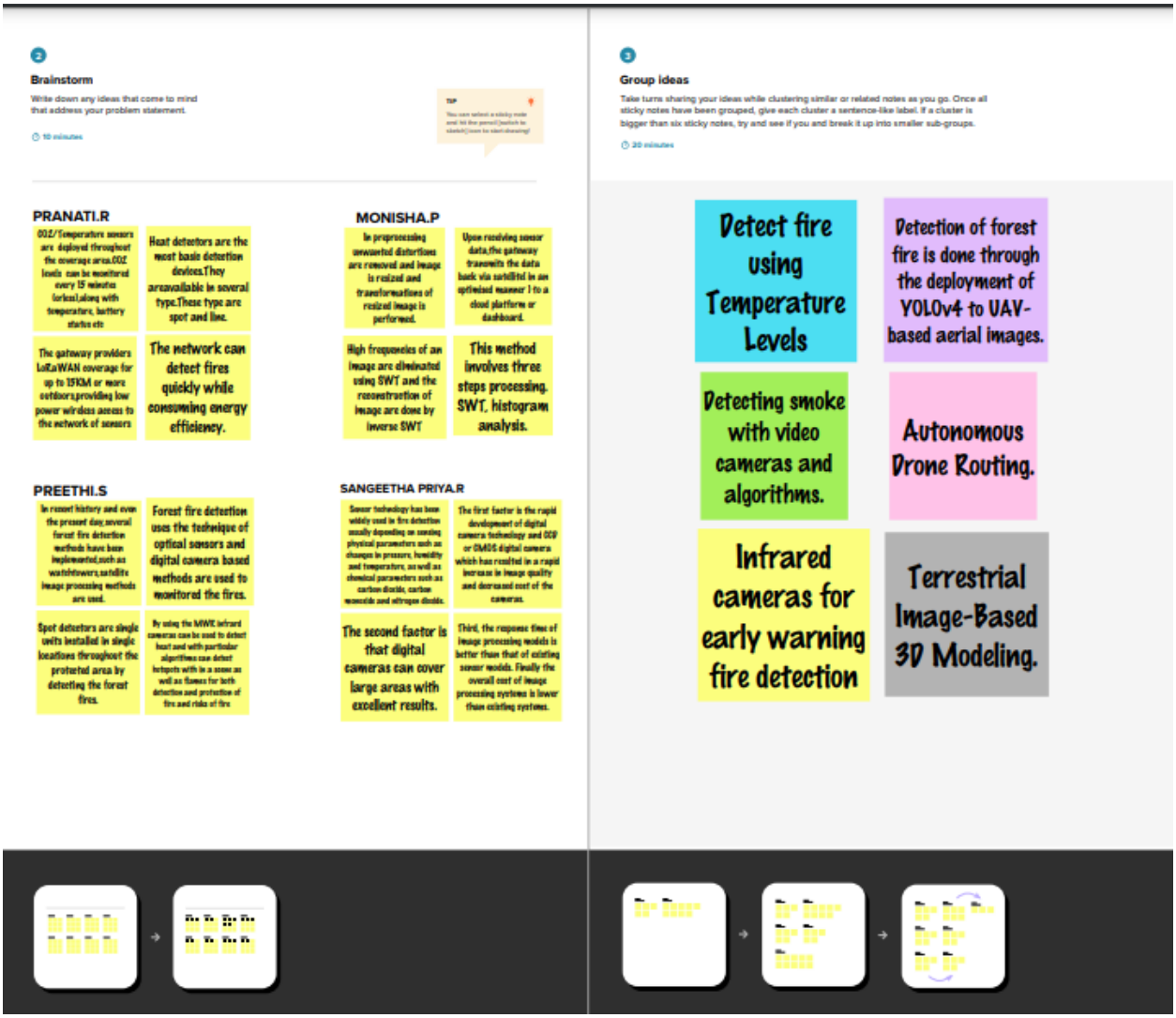
- *But this approach was slow because the fire may have spread in the largeareas and caused a lot of damage before the rescue team arrived.*
- *Since it's impossible to place a man inevery part of a forest,it's important to have monitoring devices in certainareas so we can keep an eye on the forest.*
- *Both watching towers and satellite images failed to detect the presenceof a fire earlyon, which resulted in more damage being done by the fire.*
- *Predictive analytics based on these insights are becoming increasingly effective in detecting mitigating and preventing fires.*

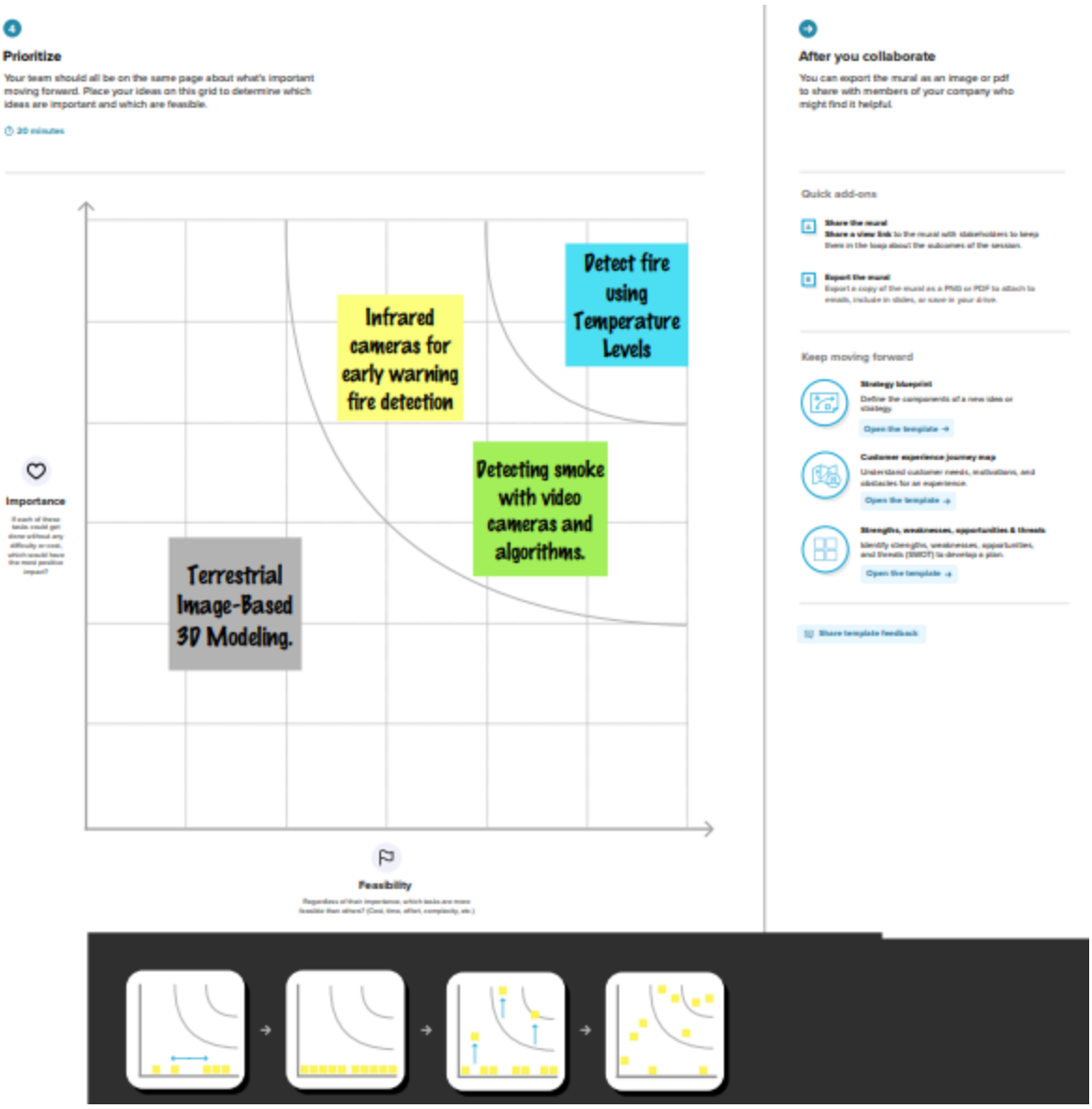
# *3.IDEATION AND PROPOSED SOLUTION*

## *3.1 Empathy Map Canvas*



# *3.2 Ideation and Brainstorming*


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
PRANATI.R
CO2/Temperature sensors are deployed throughout the coverage area.CO2 levels can be monitored every 15 minutes (orless),along with temperature, battery status etc
Heat detectors are the most basic detection devices.They areavailable in several type.These type are spot and line.
The gateway providers LoRaWAN coverage for up to 15KM or more outdoors,providing low power wireless access to the network of sensors
The network can detect fires quickly while consuming energy efficiency.
MONISHA.P
In preprocessing unwanted distortions are removed and image is resized and transformations of resized image is performed.
Upon resolving sensor data,the gateway transmits the data back via satellite in an optimized manner 1 to a cloud platform or dashboard.
High frequencies of an image are eliminated using SWT and the reconstruction of image are done by inverse SWT
This method involves three steps processing. SWT, histogram analysis.
PREETHI.S
In recent history and even the present day, several forest fire detection methods have been implemented,such as watchtowers,satellite image processing methods are used.
Forest fire detection uses the technique of optical sensors and digital camera based methods are used to monitored the fires.
Spot detectors are single units installed in single locations throughout the protected area by detecting the forest fires.
By using the MWIR infrared cameras can be used to detect heat and with particular algorithms can detect hotspots with in a scene as well as flames for both detection and protection of fire and risks of fire
SANGEETHA PRIYA.R
Sensor technology has been widely used in fire detection usually depending on sensing physical parameters such as changes in pressure, humidity and temperature, as well as chemical parameters such as carbon dioxide, carbon monoxide and nitrogen dioxide.
The first factor is the rapid development of digital camera technology and CCD or CMOS digital camera which has resulted in a rapid increase in image quality and decreased cost of the cameras.
The second factor is that digital cameras can cover large areas with excellent results.
Third, the response time of image processing models is better than that of existing sensor models. Finally the overall cost of image processing systems is lower than existing systems.
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
Detect fire using Temperature Levels
Detection of forest fire is done through the deployment of YOLOv4 to UAV-based aerial images.
Detecting smoke with video cameras and algorithms.
Autonomous Drone Routing.
Infrared cameras for early warning fire detection
Terrestrial Image-Based 3D Modeling.


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
Infrared cameras for early warning fire detection
Detect fire using Temperature Levels
Detecting smoke with video cameras and algorithms.
Terrestrial Image-Based 3D Modeling.
Importance
Feasibility
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
Share the mural
Export the mural
Keep moving forward
Strategy blueprint
Open the template
Customer experience journey map
Open the template
Strengths, weaknesses, opportunities & threats
Open the template
Share template feedback

## 3.3 Proposed Solution

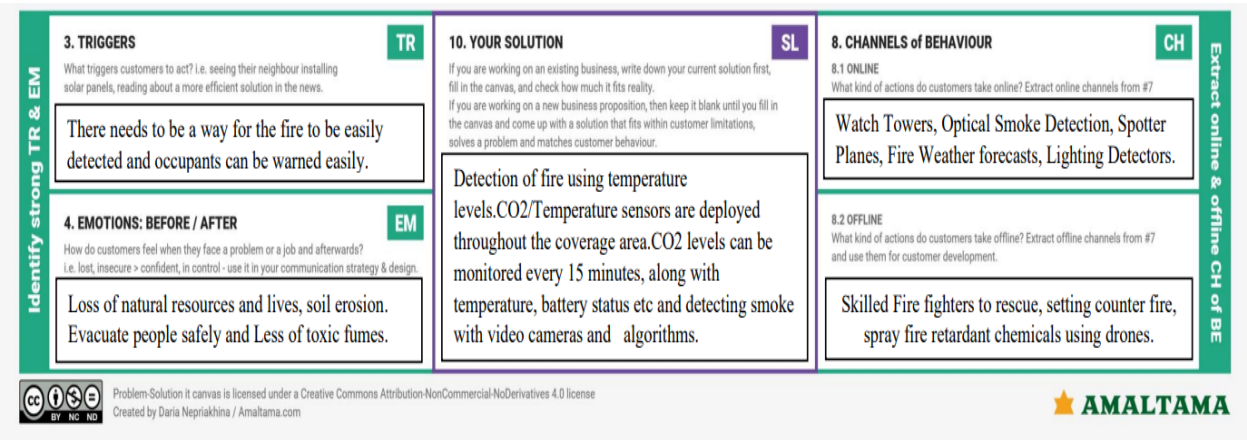
| *S.No.* | *Parameter* | *Description* |
|---|---|---|
| *1.* | *Problem Statement (Problem to be solved)* | *Over the last few decades, forest fires are increased due to deforestation and global warming. Many trees and animals in the forest are affected by forest fires. Technology can be efficiently utilized to solve this problem. Forest fire detection is inevitable forforest fire management.* |
| *2.* | *Idea / Solution description* | *Modern fire protection systems are comprised of three main components — fire detection, alarms and notifications, and suppression, all of which must function together to provide the necessary fire protection for a given building. Designing a fire alarm and notification system requires an integrated approach that includes a comprehensive analysis of the entire fire protection system. This analysis is necessary to gain a thorough understanding of how all the main components of the overall fire protection system will work together. This analysis needs to be conducted before the system is installed.* |
| *3.* | *Novelty / Uniqueness* | *The novelty of system is real-time monitoring, early prediction, validation through UAV and fire confirmation using image processing. The proposed system presents highertrue fire detection rate of about 95-98 percent.* |

| | | |
|---|---|---|
| ***4.*** | ***Social Impact / CustomerSatisfacti on*** | ***Timely information aboutthe 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. Monitoring of the potential risk areas and an early detection of fire can significantly shorten the reaction time and also reduce the potential damage caused by the forest fire.*** |

| | | |
|---|---|---|
| ***5.*** | ***Business Model (Revenue Model)*** | ***Aspirating smoke detectors continuously sample air to provide early warnings of fire hazards, helping detect threats before they escalate. Some devices provide multi-level warnings and are equipped with wide- ranging sensitivity to identify eventhe most negligible amounts of smoke, helping to prevent smalls fire from taking hold and causing widespread damage.***<br><br>***Unlike traditional detection technology – which is largely passive, waiting for smoke to reach sensors – aspirating devices are designed to sample and test air near the most likely sources of fires throughout a building. Aspirating smoke devices can be positioned in hard-to-monitor places, such as ceilings, air grilles and openings, or within critical spaces, including operating and patient rooms. Early detection technologies can also draw air from targeted locations back to a central system that continuously monitors for trace amounts of smoke.*** |

| | | |
|---|---|---|
| ***6.*** | ***Scalability ofthe Solution*** | ***Changes in the use or occupancy of abuilding can resultin compliance issuesand a fire alarm system that no longer provides sufficient protection. If future changes are anticipated, fire safety engineers can design a fire alarm system with this in mind, providing a flexible infrastructure that includes the proper wire size and additional circuits distributed in a way that accommodates futuregrowth and change.***<br>***Perhaps one of the most compelling reasons to design a fire alarm system that goes above and beyond the minimum requirements from the start is the fact that fire codes and other applicable regulations can and do change.And, changes that are made retroactively can trigger potentially very expensive alterations in a fire alarm system. This is also why it is so important to work with highly qualified fire safety engineers who can anticipate coming changes andproactively design your system to meet new requirements.*** |

# 3.4 Problem Solution Fit

# 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* | *Video surveillance start* | *Start surveillance throughremote control* |
| *FR-2* | *Forest monitoring* | *Continuous monitoring through camera* |
| *FR-3* | *Detect fire* | *Fire is detectedthrough CNN model* |
| *FR-4* | *Alert* | *Alert the forest officials through message* |

## 4.2 Non-functional Requirements:

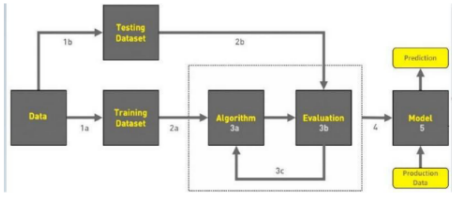
**Following are the non-functional requirements of the proposed solution.**

| *FR No.* | *Functional Requirement(Epic)* | *Sub Requirement (Story / Sub-Task)* |
|---|---|---|
| *FR-1* | *Reliability* | *Model is safe to install* |
| *FR-2* | *Security* | *More secure environment* |
| *FR-3* | *Availability* | *Build model is available all the time* |

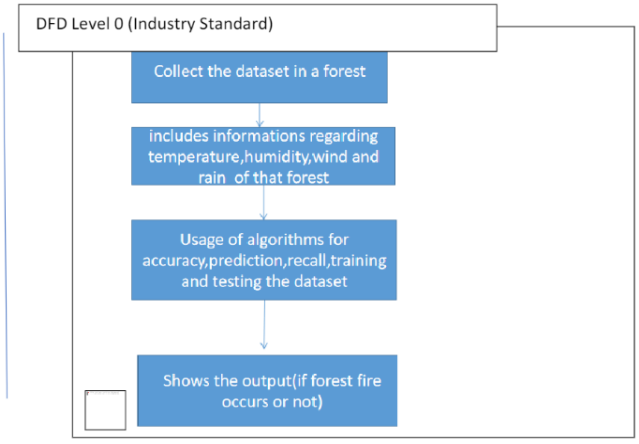
| | | |
|---|---|---|
| ***FR-4*** | ***Performance*** | ***Model will achieve high accuracy*** |

# 5. PROJECT DESIGN

## 5.1Data Flow Diagrams




1. COLLECT DATA
2. EVALUATE DATA SET
3. IMPLEMENT ALGORITHMS
4. EVALUATE THE ACCURACY OF EACH ALGORITHMS
5. DISPLAY RESULTS




## *5.2 Solution and Technical Architecture*

## *Solution Architecture*

1. ***This Solution Architecture involves four stages.***
    a. ***Input Image***
    b. ***Region Proposal***
    c. ***Feature extraction &classification***
    d. ***Output detection result***

***Step 1: We get the input image and discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are***

***mapped out.***

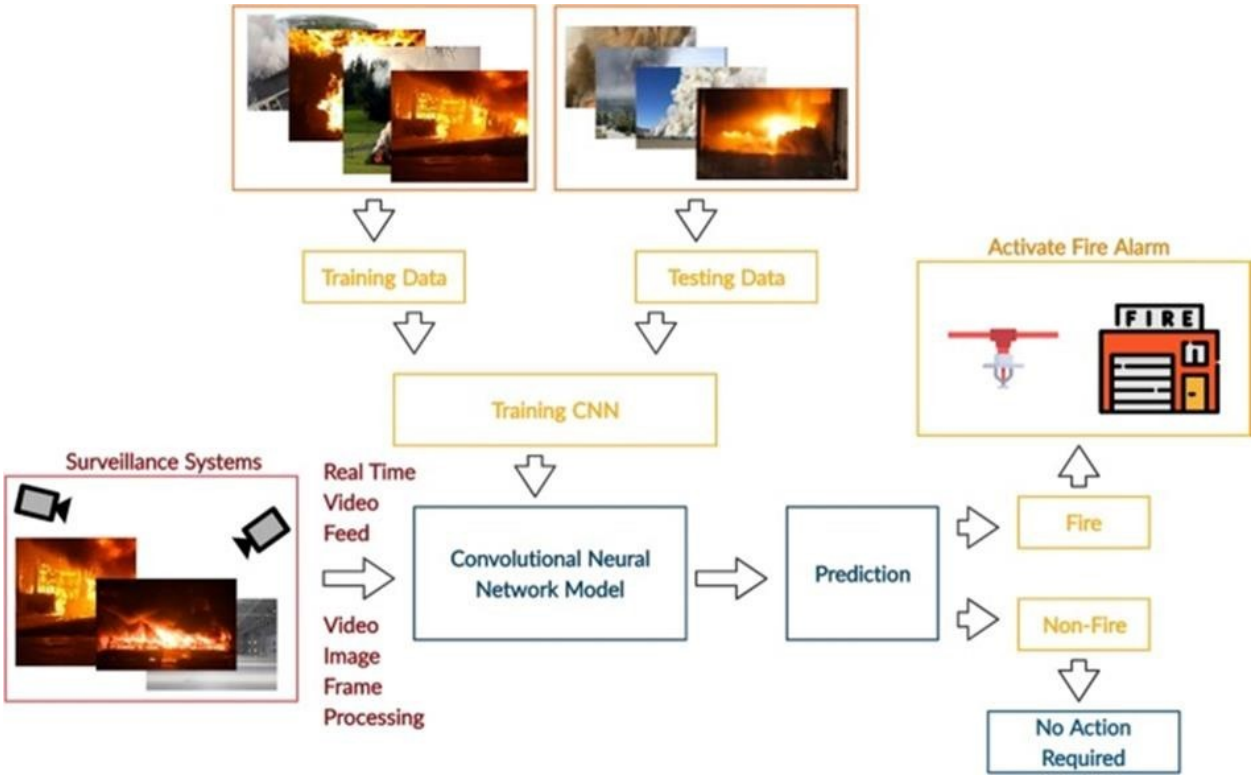
***Step 2: The second part of this step will involve the Rectified Linear Unit or ReLU. We will cover ReLU layers and explore how linearity functions in the context of Convolutional Neural Networks.***
***Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.***

***Step 3-Pooling: In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.***

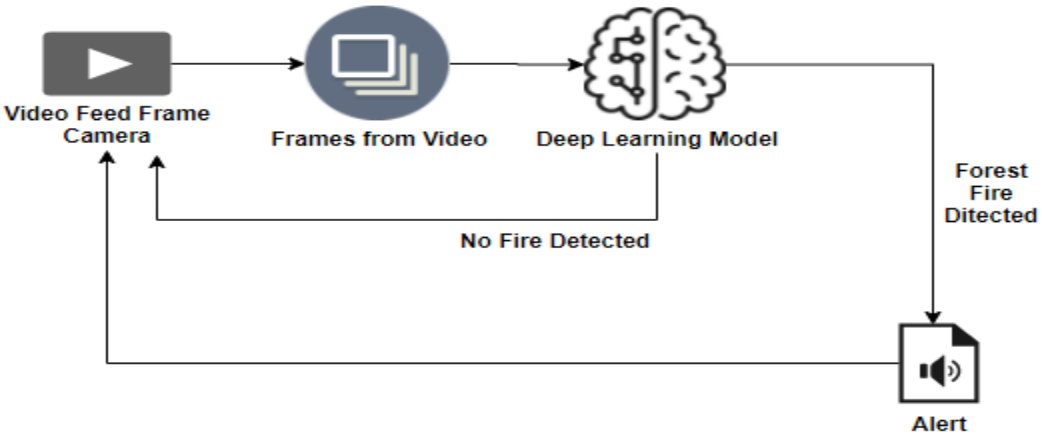
***Step 4 -Flattening: This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional NeuralNetworks.***

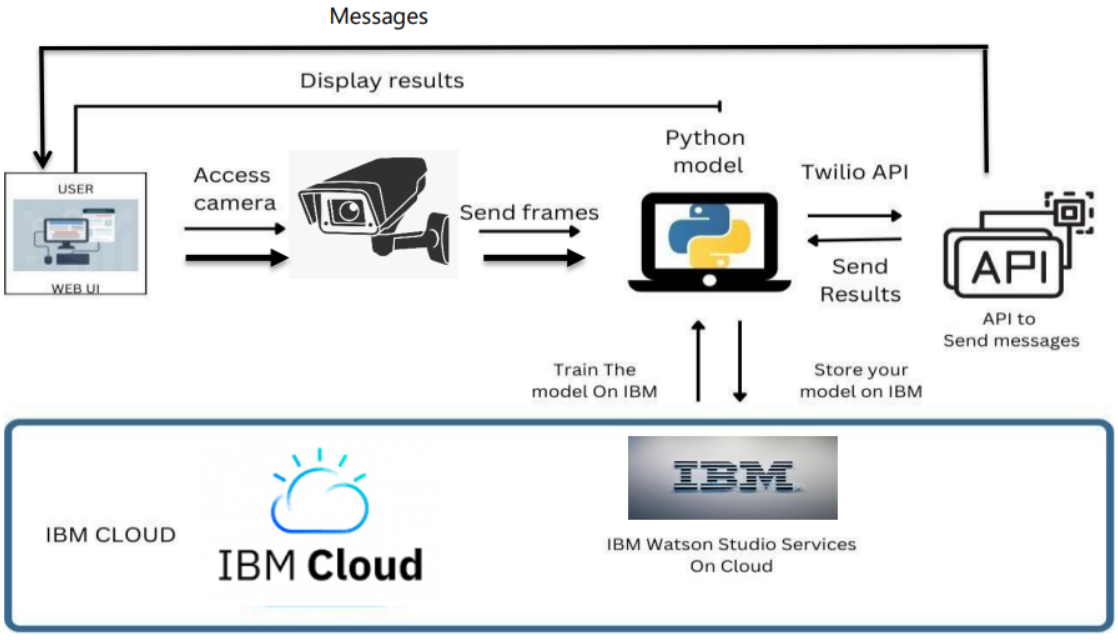
***Step 5-FullConnection: In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how***

***Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.***



# Technology Architecture



## *Table 1: Components and 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 intoFrames* | *Frame Converter* |
| *4.* | *Feeding the Model* | *The Frames are sent to the Deeplearning model* | *Our Model* |
| *5.* | *Dataset* | *Using Test set and Train set , train the model* | *Data set from CloudStorage , Database* |
| *6.* | *Cloud Database* | *The model is trained in the cloud more precise with* | *IBM Cloud ant,Python Flask.* |

| | | | |
|---|---|---|---|
| | | ***detections later images can be added.*** | |
| ***7.*** | ***Infrastructure (Server / Cloud), API*** | ***Application Deployment on Local System / Cloud Local ,CloudServer Configuration , Twilio API to sendmessages*** | ***Java/python, React.Js, JavaScript, HTML , CSS ,IBM Cloud,OPEN CV,***<br>***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 Controlis used*** | ***e.g. SHA-256, Encryption s, IAM Controls, OWASPetc.*** |
| ***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 acrossservers* | *IBM loadbalancer* |
| *5.* | *Performance* | *Enhance the performance by using IBM CDN* | *IBM Content Delivery Network* |

## *5.3 User Stories*

**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 |
| | | 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 |
| | | USN-5 | Identify accuracy,precision,recall of each algorithms | These values are important for obtaining the right output | High | Sprint-3 |
| | | USN-6 | Outputs from each algorithm are obtained | It is highly used to predict the effect and to take precautionary measures. | High | Sprint-4 |

# 6.PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | DATA COLLECTION | USN-1 | Data collected by sensors aboard orbiting satellites, carried aboard aircraft, or installed on the ground provide a wealth of data that can be used to assess conditions before a burn and track the movement of a wildfire in near real-time. | 10 | High | Pranati<br>Monisha<br>Preethi<br>Sangeetha Priya |
| Sprint-1 | IMAGE PREPROCESSING | USN-2 | Image processing-Image processing technique automatically detect forest fires around the world by using infrared(IR) images sourced from satellites and CNN used for image recognition and tasks that involve the processing of pixel data. | 7 | Medium | Pranati<br>Monisha<br>Preethi<br>Sangeetha Priya |
| Sprint-2 | TRAINING AND TESTING | USN-3 | The model is trained for detecting the fire by training with real time work and the testing is done according the accuracy of the model | 10 | high | Pranati<br>Monisha<br>Preethi<br>Sangeetha Priya |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-3 | REVIEWING THE MODEL | USN-4 | The main task is to check that the model is efficient to work in real time to ensure there is no error in the model | 7 | Medium | Pranati<br>Monisha<br>Preethi<br>Sangeetha Priya |
| Sprint-4 | IMPLEMENTATION | USN-5 | After completing every step the model is implemented on the forest and the quick responses is collected from forest organization | 10 | High | Pranati<br>Monisha<br>Preethi<br>Sangeetha Priya |

## *6.2 Sprint Delivery Schedule*

**Project Tracker, Velocity & Burndown Chart:**

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 8 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 7 | 08 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 8 | 15 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 7 | 20 Nov 2022 |

**Velocity:**

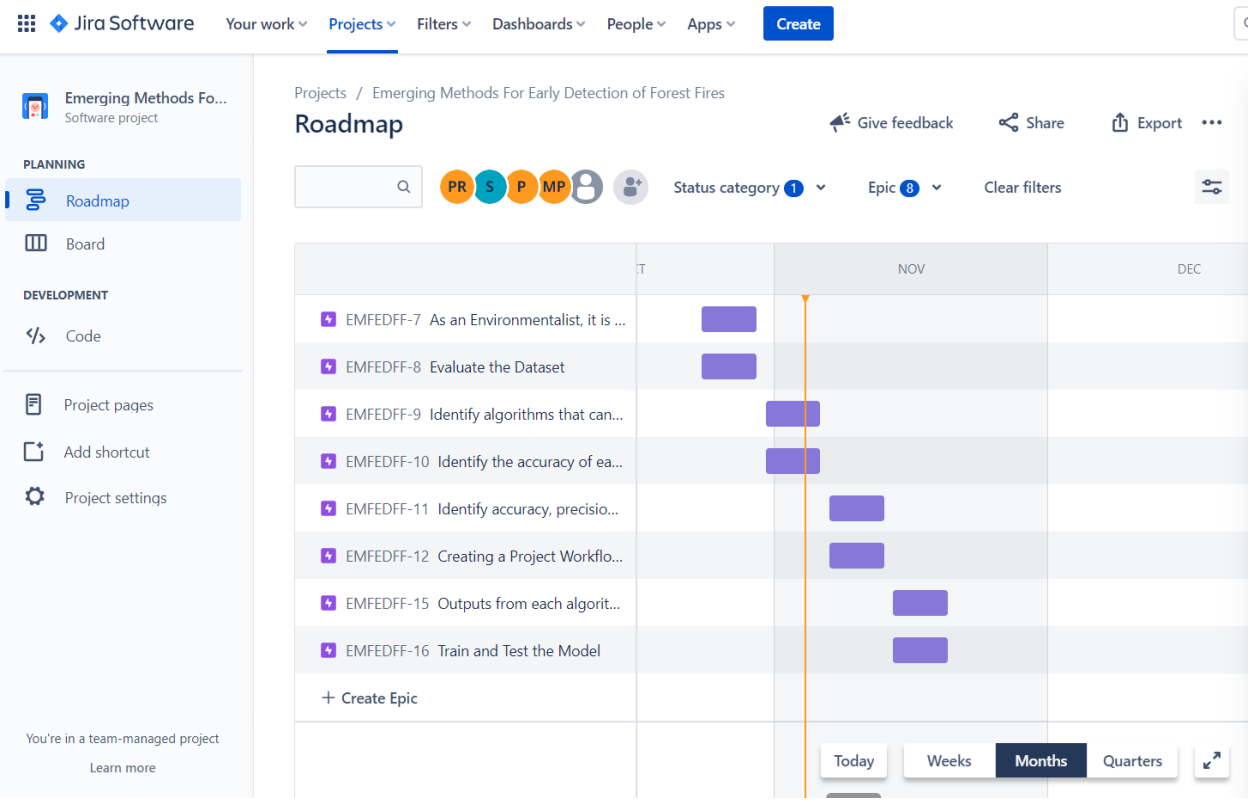
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = 7/10 = 0.7$$

**Burndown Chart:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

# *6.3 Reports from JIRA*



# *7.CODING AND SOLUTION*

## *7.1 Feature 1*

- ***Language used: Python***
- ***Tools/IDE:Google Colab***

In [107…

```
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=True).
```

Data Collection

In [108…

```
!unzip '/content/drive/MyDrive/archive.zip'
```

```
Archive:  /content/drive/MyDrive/archive.zip
replace Dataset/Dataset/test_set/forest/0.48007200_1530881924_final_forest.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

Image Pre-processing

In [109…

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   rotation_range=180,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
train = train_datagen.flow_from_directory('/content/Dataset/Dataset/test_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')
test = train_datagen.flow_from_directory('/content/Dataset/Dataset/train_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')
```

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

Sprint 2

In [110…

```
#Model Building
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPooling2D,Dense,Flatten
import warnings
warnings.filterwarnings('ignore')
```

In [111…

```
#Initializing the model and adding CNN and Dense layers
model = Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=256,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      896

max_pooling2d (MaxPooling2D  (None, 63, 63, 32)        0
)

flatten (Flatten)            (None, 127008)            0

dense (Dense)                (None, 256)               32514304

dense_1 (Dense)              (None, 1)                 257

=================================================================
Total params: 32,515,457
Trainable params: 32,515,457
Non-trainable params: 0
_________________________________________________________________
```

In [112…

```
# Compiling the Model
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy','mse'])
```

In [113…

```
#Training the model
y = model.fit_generator(train,steps_per_epoch=14,epochs=15,validation_data=test,validation_steps=4)
```

```
Epoch 1/15
 4/14 [=======>......................] - ETA: 27s - loss: 4.0799 - accuracy: 0.5537 - mse: 0.3706
```

```
14/14 [==============================] - 27s 1s/step - loss: 4.0799 - accuracy: 0.5537 - mse: 0.3706 - val_loss: 6.6469 - val_accuracy: 0.6562 - val_mse: 0.3400
```

In [114…

```
#Saving the model
model.save('ffd_model.h5')
```

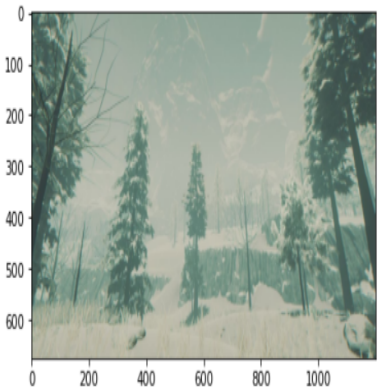
In [116…

```
#Testing the model
from keras.models import load_model
import cv2
import numpy as np
from PIL import Image
from keras.utils import img_to_array
model = load_model('/content/ffd_model.h5')
def prediction(img_path):
    i = cv2.imread(img_path)
    i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
    img = Image.open(img_path)
    img = img.resize((128,128))
    x = img_to_array(img)
    x = np.expand_dims(x,axis=0)
    pred = model.predict(x)
    plt.imshow(i)
    print("%s"%("FOREST FIRE DETECTED! SMS SENT!" if pred==[[1.]] else "NO FOREST FIRE DETECTED"))
```

In [117…

```
prediction(r'/content/Dataset/Dataset/test_set/forest/1200px_Mountainarea.jpg')
```

```
1/1 [==============================] - 0s 182ms/step
NO FOREST FIRE DETECTED
```



In [118…

```
prediction(r'/content/Dataset/Dataset/test_set/with fire/Fire_2_696x392.jpg')
```

```
1/1 [==============================] - 0s 61ms/step
FOREST FIRE DETECTED! SMS SENT!
```


Model Deployment in IBM Cloud

In [119…

```
#Converting .h5 to tar format
!tar -zcvf forest_fire_detection.tgz ffd_model.h5
```

```
ffd_model.h5
```

In [ ]:

```
!pip install watson-machine-learning-client
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: watson-machine-learning-client in /usr/local/lib/python3.7/dist-packages (1.0.391)
Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.26.12)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2.28.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (4.64.1)
```

# *7.2 Feature 2*

Model Deployment in IBM Cloud

In [119…

```
#Converting .h5 to tar format
!tar -zcvf forest_fire_detection.tgz ffd_model.h5
```

```
ffd_model.h5
```

In [ ]:

```
!pip install watson-machine-learning-client
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: watson-machine-learning-client in /usr/local/lib/python3.7/dist-packages (1.0.391)
Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.26.12)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2.28.1)
```

```
1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (1.21.6)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (2022.6)
```

In [ ]:

```
!pip install ibm_watson_machine_learning
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting ibm_watson_machine_learning
  Downloading ibm_watson_machine_learning-1.0.257-py3-none-any.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 6.5 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (4.13.0)
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.8.10)
Collecting ibm-cos-sdk==2.7.*
  Downloading ibm-cos-sdk-2.7.0.tar.gz (51 kB)
     |████████████████████████████████| 51 kB 498 kB/s
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.26.12)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.3.5)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2.28.1)
Collecting ibm-cos-sdk-core==2.7.0
  Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)
     |████████████████████████████████| 824 kB 39.4 MB/s
Collecting ibm-cos-sdk-s3transfer==2.7.0
```

```
Successfully installed docutils-0.15.2 ibm-cos-sdk-2.7.0 ibm-cos-sdk-core-2.7.0 ibm-cos-sdk-s3transfer-2.7.0 ibm-watson-machine-learning-1.0.257
```

In [ ]:
```
#Connecting to IBM Cloud from Notebook
from ibm_watson_machine_learning import APIClient
credentials = {
    'url':'https://us-south.ml.cloud.ibm.com',
    'apikey':"hwPqBMWeHLVUWozQrsf8OwqAZUlLPITGwY4gMKcMBpVF"
}
Client = APIClient(credentials)
```

```
Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future release. Use Python 3.9 framework instead.
```

In [ ]:
```
Client
```

Out[ ]:

In [ ]:
```
Client.spaces.get_details()
```

Out[ ]:
```
{'resources': [{'entity': {'description': '',
    'name': 'forest fires',
    'scope': {'bss_account_id': '29b20e18cb82499ca758899f43447824'},
    'stage': {'production': False},
    'status': {'state': 'active'},
    'storage': {'properties': {'bucket_name': '0bc342a6-1621-4eeb-ba26-b44087ff24d6',
      'bucket_region': 'us-south',
      'credentials': {'admin': {'access key id': 'e6abbbbf7099406aa52ed55a4701c60f',
```

In [ ]:
```
Client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  --------------------  -------------------------
ID                                    NAME                  CREATED
33e29996-acf0-40b6-8882-c6a1506201c7  forest fires project  2022-11-10T14:58:43.819Z
1efdc4c1-392d-4714-89f6-6db05f965f7a  forest fires          2022-11-10T14:56:08.672Z
------------------------------------  --------------------  -------------------------
```

In [ ]:
```
space_uid = '33e29996-acf0-40b6-8882-c6a1506201c7' #Space User ID
space_uid
```

Out[ ]:
```
'33e29996-acf0-40b6-8882-c6a1506201c7'
```

In [ ]:
```
#Setting created deployment space as default
Client.set.default_space(space_uid)
```

Out[ ]:
```
'SUCCESS'
```

In [ ]:
```
#Seeing tensorflow asset_id
Client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
```

In [ ]:
```
software_space_uid = Client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

Out[ ]:
```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

In [ ]:
```
model_details = Client.repository.store_model(model="/content/forest_fire_detection.tgz",meta_props={
    Client.repository.ModelMetaNames.NAME:"forest fires project",
    Client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    Client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

In [ ]:
```
model_details
```

Out[ ]:
```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_2.7'},
 'metadata': {'created_at': '2022-11-11T07:11:08.554Z',
  'id': 'cd777ca7-c414-4c5d-9fe2-6c9828275c1b',
  'modified_at': '2022-11-11T07:11:30.184Z',
  'name': 'forest fires project',
  'owner': 'IBMid-665002KXQR',
  'resource_key': '2bea2454-f5f8-49d9-b224-07068c1dd165',
  'space_id': '33e29996-acf0-40b6-8882-c6a1506201c7'},
 'system': {'warnings': []}}
```

In [ ]:
```
model_id = Client.repository.get_model_uid(model_details)
model_id
```

Out[ ]:
```
'cd777ca7-c414-4c5d-9fe2-6c9828275c1b'
```

In [ ]:
```
#Downloading the model from IBM Cloud
Client.repository.download(model_id,'ffd_model.tgz')
```

```
Successfully saved model content to file: 'ffd_model.tgz'
```

Out[ ]:
```
'/content/ffd_model.tgz'
```

In [ ]:
```
!pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting twilio
  Downloading twilio-7.15.2-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 8.8 MB/s
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.28.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.26.12)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.2
```

Video Analysis

In [86]:
```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: twilio in /usr/local/lib/python3.7/dist-packages (7.15.2)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.28.1)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.6.0)
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: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.26.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
```

In [87]:
```
pip install playsound
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: playsound in /usr/local/lib/python3.7/dist-packages (1.2.2)
```

In [88]:
```
#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
```

In [89]:
```
#load the saved model
model = load_model(r'/content/ffd_model.h5')
#define video
video = cv2.VideoCapture('/content/demo.mp4')
#define the features
name = ['forest','with forest']
```

In [122…
```
account_sid = 'ACe316bbd6e26b5f3fba3c0798903db32a'
auth_token = '1cedc0d00f2354840f10ba5810c6c7fd'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
      body='Forest fire is detected , stay alert',
      from_='+18316535983',
      to='+918838258974'
     )

print(message.sid)
print("Fire Detected")
print("SMS Sent")
```

```
SM50c6779b73363d18e28dac365d5919bf
Fire Detected
SMS Sent
```

# *8.TESTING*

## *8.1 Test Cases*

| Test case ID | Feature Type | Component | Test Scenario | Steps To Execute | Test Data | Expected Result | Actual Result | Status | BUG ID | Executed By |
|---|---|---|---|---|---|---|---|---|---|---|
| OP_RT_001 | Functional | Page | Check if user can upload their file | The sensor senses the fire | Sample 1.png | The input image should be uploaded to the application successfully | Working as expected | PASS | | R.Pranati<br>P.Monisha |
| OP_RT_002 | Functional | Page | Check if user cannot upload unsupported files | 1) The sensor senses the fire<br>2)checks with the pre-uploads images | installer.exe | The application should not allow user to select a non image file | User is able to upload any file | FAIL | BUG_HP_002 | S.Preethi<br>R.Sangeetha Priya |
| OP_RT_003 | Functional | Page | Checks whether the page redirects to the result page to the given output | 1) The sensor senses the fire<br>2)checks with the pre-uploaded images<br>3)checks if there is fire detection | Sample 1.png | The page should redirect to the results page | Working as expected | PASS | | R.Pranati |
| MB_RT_001 | Functional | Backend | Checks if all the routes are working properly | 1) The sensor senses the fire<br>2)checks with the pre-uploaded images<br>3)checks if there is fire detection | Sample 1.png | All the routes should properly work | Working as expected | PASS | | P.Monisha |
| N_DC_001 | Functional | Model | Checks whether the model can handle various image sizes | 1) Open the page in a specific device<br>2) Upload the input image<br>3) Repeat the above steps with different input | Sample 1.png<br>Sample 1 XS.png<br>Sample 1 XL.png | The model should rescale the image and predict the results | Working as expected | PASS | | R.Sangeetha Priya |
| N_DC_002 | Functional | Model | Check if the model predicts the digit | 1) Open the page<br>2) Select the input images | Sample 1.png | The model should predict the number | Working as expected | PASS | | R.Sangeetha Priya<br>S.Preethi |
| N_DC_003 | Functional | Model | Check if the model can handle complex input image | 1) Open the page<br>2) Select the input images<br>3) Check the results | Complex Sample.png | The model should predict the number in the compex image | The model fails to identify the digit since the model is not built to handle such data | FAIL | BUG_M_001 | R.Pranati<br>P.Monisha |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| RL_DC_001 | Functional | Result Page | Verify the elements | 1) Open the page<br>2) Select the input image<br>3) Check if all the UI elements are displayed properly | Sample 1.png | The Result page must be displayed properly | Working as expected | PASS | | R.Pranati<br>P.Monisha |
| RL_DC_002 | Functional | Result Page | Check if that image is displayed properly | 1) Open the page<br>2) Select the input image<br>3) Check if the input image are displayed | Sample 1.png | The input image should be displayed properly | The size of the input image exceeds the display container | FAIL | BUG_RP_001 | R.Sangeetha Priya<br>S.Preethi |
| RL_DC_003 | Functional | Result Page | Checks whether the displayed prediction is accurate | 1) Open the page<br>2) Select the input image<br>3) Check if all the other predictions are displayed | Sample 1.png | The other predictions should be displayed properly | Working as expected | PASS | | **R.Pranati**<br>**R.Sangeetha Priya** |

# *8.2 User Acceptance Testing*

## 1. Purpose of Document

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing are done.

The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. The arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

## 2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 1 | 1 | 2 | 0 | 4 |
| Duplicate | 0 | 0 | 0 | 0 | 0 |
| External | 0 | 0 | 2 | 1 | 3 |
| Fixed | 4 | 2 | 4 | 1 | 11 |
| Not Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 0 | 0 | 1 | 1 |
| Totals | 5 | 3 | 9 | 4 | 21 |

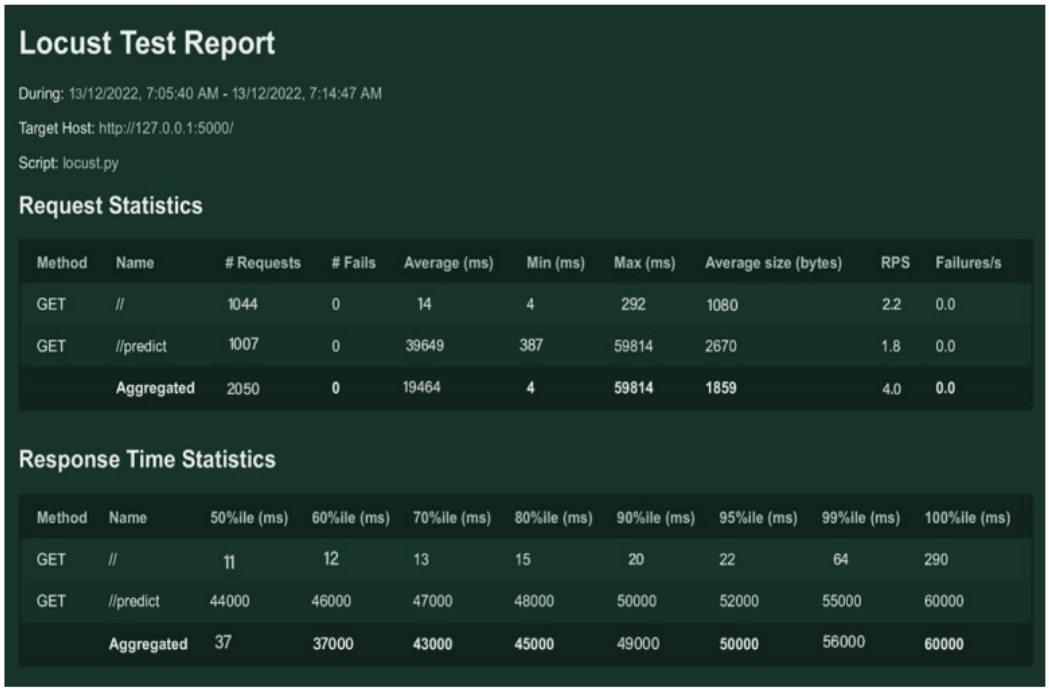
## 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Client Application | 10 | 0 | 0 | 10 |
| Security | 2 | 0 | 0 | 2 |
| Performance | 2 | 0 | 0 | 2 |
| Exception Reporting | 2 | 0 | 0 | 2 |
| Final Report Output | 3 | 0 | 0 | 3 |

# *9.Results*

## *9.1 Performance Metrics*



Locust Test Report

During: 13/12/2022, 7:05:40 AM - 13/12/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

Script: locust.py

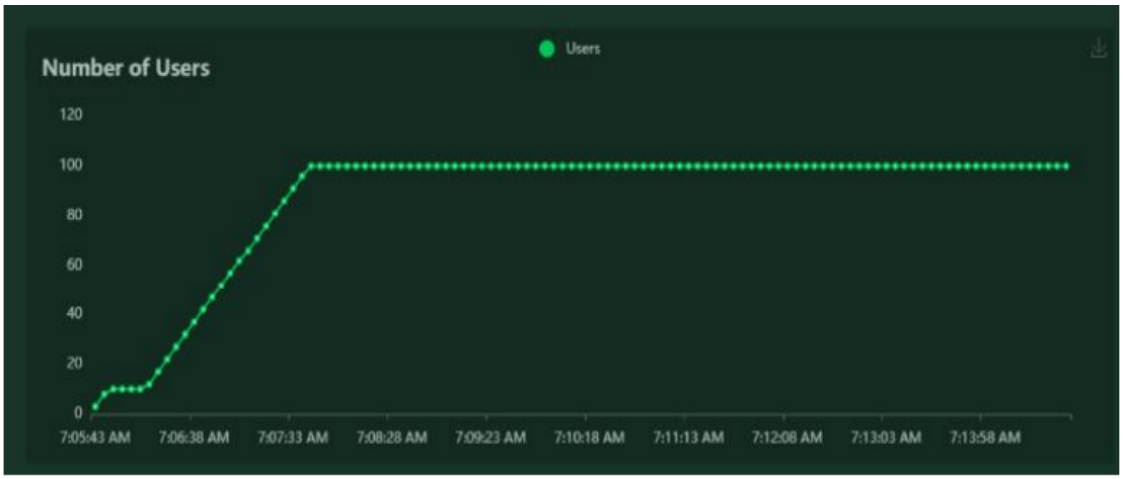
Request Statistics

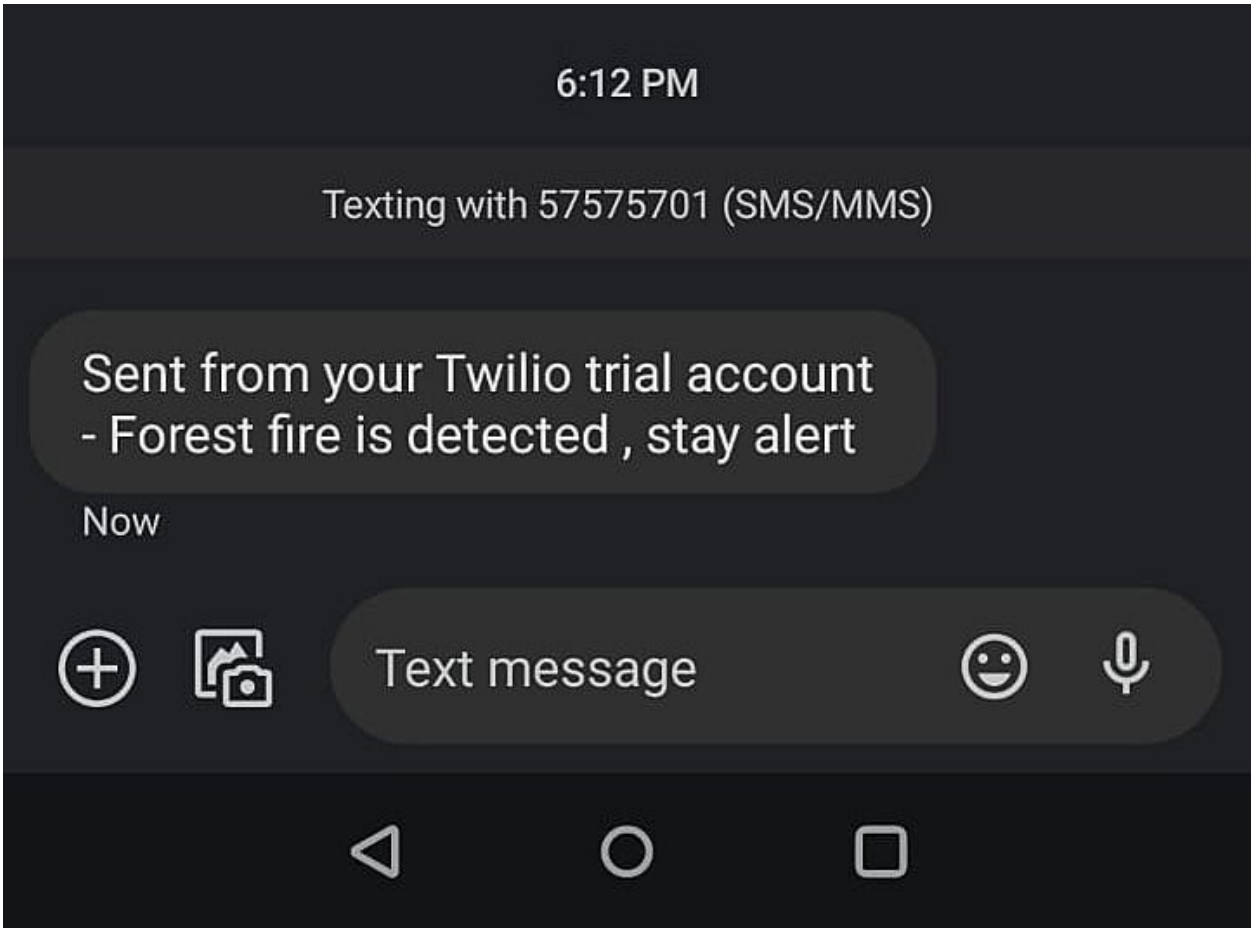
| Method | Name | # Requests | # Fails | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | RPS | Failures/s |
|---|---|---|---|---|---|---|---|---|---|
| GET | / | 1044 | 0 | 14 | 4 | 292 | 1080 | 2.2 | 0.0 |
| GET | /predict | 1007 | 0 | 39649 | 387 | 59814 | 2670 | 1.8 | 0.0 |
| | **Aggregated** | 2050 | **0** | 19464 | **4** | **59814** | **1859** | 4.0 | **0.0** |

Response Time Statistics

| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | / | 11 | 12 | 13 | 15 | 20 | 22 | 64 | 290 |
| GET | /predict | 44000 | 46000 | 47000 | 48000 | 50000 | 52000 | 55000 | 60000 |
| | **Aggregated** | 37 | **37000** | **43000** | **45000** | 49000 | **50000** | 56000 | **60000** |


Charts
Total Requests per Second
RPS
Failures/s
6
5
4
3
2
1
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Response Times (ms)
Median Response Time
95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM


Number of Users
Users
120
100
80
60
40
20
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM


6:12 PM
Texting with 57575701 (SMS/MMS)
Sent from your Twilio trial account
- Forest fire is detected , stay alert
Now
Text message

## *10. ADVANTAGES AND DISADVANTAGES*

### *ADVANTAGES:*

1. ***The proposed model can be used in combination with a nightcamera and a thermal camera in a forest to identify tiny fire signs.***
2. ***More datasets and images can be used to train for a more accurateoutcome when detecting flame destruction ability.***
3. ***The model can be implemented in mobile***
4. ***applications for camping experience enthusiasts.***

### *DISADVANTAGES:*

1. ***The model works for limited information.***
2. ***The accuracy is low becauseto the limited quantity/quality of photos in the dataset, but this may easily be increased by changing the dataset.***
3. ***The small amount of fire amount detection can also cause to trigger the alarm.***

## *APPLICATIONS:*

1. ***It will contribute to surveillance technology that improves the accuracy and predictability of fire detection.***
2. ***able to detect the fire forestmore precisely, as well as some forest plants and wildlife.***
3. ***Detect the amount of dangers that should be treated and those that should not. extra assistance in contacting fire fighters for assistance system.***

## *11.CONCLUSION*

***Forest fires are a major cause of rain forest and savanna degradation. This model will aid in minimizing destruction by anticipating it to the system, allowing individuals to react more quickly and prevent it. The proposed methodology would deconstruct the threat to the environment by converting the image collected into signals that will trigger an alarm. This system transmits video images to a model,which recognizes them and determines whether to send a threat alert or not. The model extracts data from video feeds and defines image***

***processing into RGB data for signal response modelling.***

## *12.FUTURE SCOPE*

***SThe availability of fire-fighting technology brings us one step closer to new AI for detection and security in the forest and at home. With the addition of a motion sensor, the technology can simply expand to compact decision-making with the addition of new software and hardware. The system is utilized as a drone and surveillance system UAV to expand the surveillance area and detect heat signatures in order to identify human from fire plasma signatures.***

# *13.APPENDIX*

## *13.1 Source Code*

In [86]:
```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: twilio in /usr/local/lib/python3.7/dist-packages (7.15.2)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.28.1)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.6.0)
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: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.26.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
```

In [87]:
```
pip install playsound
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: playsound in /usr/local/lib/python3.7/dist-packages (1.2.2)
```

In [88]:
```
#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
```

In [89]:
```
#load the saved model
model = load_model(r'/content/ffd_model.h5')
#define video
video = cv2.VideoCapture('/content/demo.mp4')
#define the features
name = ['forest','with forest']
```

In [122…
```
account_sid = 'ACe316bbd6e26b5f3fba3c0798903db32a'
auth_token = '1cedc0d00f2354840f10ba5810c6c7fd'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
     body='Forest fire is detected , stay alert',
     from_='+18316535983',
     to='+918838258974'
     )

print(message.sid)
print("Fire Detected")
print("SMS Sent")
```

```
SM50c6779b73363d18e28dac365d5919bf
Fire Detected
SMS Sent
```

## *13.2 GitHub & Project Demo Link*

***GitHub Link:https://github.com/IBM-EPBL/IBM-Project-2915-1658486320.git***

***Project Demo Link:https://uploadnow.io/f/4vP0hNy***