

NALAIYA THIRAN - IBM PROJECT REPORT
(19IT410T Professional Readiness for Innovation, Employability and Entrepreneurship)
ON
EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRE

Submitted by

TEAM ID: PNT2022TMID23374

PRANAV RAJ N K (113219031109)

GANESH S (113219031042)

PRAVIN KUMAR R(113219031112)

SELVAKUMAR R(113219031134)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

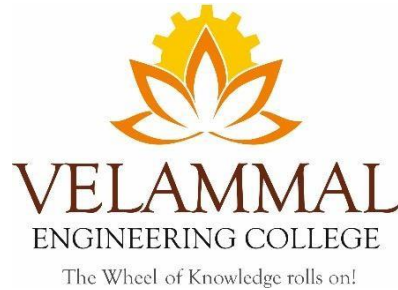


VELAMMAL ENGINEERING COLLEGE, CHENNAI-66.

(An Autonomous Institution, Affiliated to Anna University, Chennai)

VELAMMAL ENGINEERING COLLEGE CHENNAI -66

(An Autonomous Institution, Affiliated to Anna University, Chennai)



BONAFIDE CERTIFICATE

Certified that this NALAIYA THIRAN – IBM PROJECT REPORT “**EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES**” is the Bonafide work of “PRANAV RAJ N K (113219031109), GANESH S(113219031042), SELVAKUMAR R (113219031134), and PRAVIN KUMAR R(113219031112)” carried out in “PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP (NALAIYA THIRAN-IBM PROJECT)” during the Academic Year 2022-2023.

FACULTY EVALUATOR

DR. S.RAJALAKSHMI

Assistant Professor

Dept. of Information Technology
Velammal Engineering College
Chennai-600 066

HEAD OF THE DEPARTMENT

DR. B.MURUGESHWARI

Professor and Head

Dept. of Computer Science and Engineering
Velammal Engineering College
Chennai-600 066

CONTENTS

1. INTRODUCTION

Project Overview

Purpose

2. LITERATURE SURVEY

Existing problem

References

Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

Ideation & Brainstorming

Proposed Solution

Problem Solution fit

4. REQUIREMENT ANALYSIS

Functional requirement

Non-Functional requirements

5. PROJECT DESIGN

Data Flow Diagrams

Solution & Technical Architecture

User Stories

6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Sprint Delivery Schedule

Reports from JIRA

7. CODING & SOLUTIONING

Feature 1

Feature 2

8. TESTING

9. Test Cases

User Acceptance Testing

10. RESULTS

Performance Metrics

11. ADVANTAGES & DISADVANTAGES

12. CONCLUSION

13. FUTURE SCOPE

14. APPENDIX

Source Code

GitHub & Project Demo Link

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 mother nature 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 forest fires 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 estimated 340,000 premature deaths from respiratory and cardiovascular issues attributed to wildfire smoke.

The increasing frequency and severity of wildfires pose a growing threat 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 05/1731-35 Volume 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.

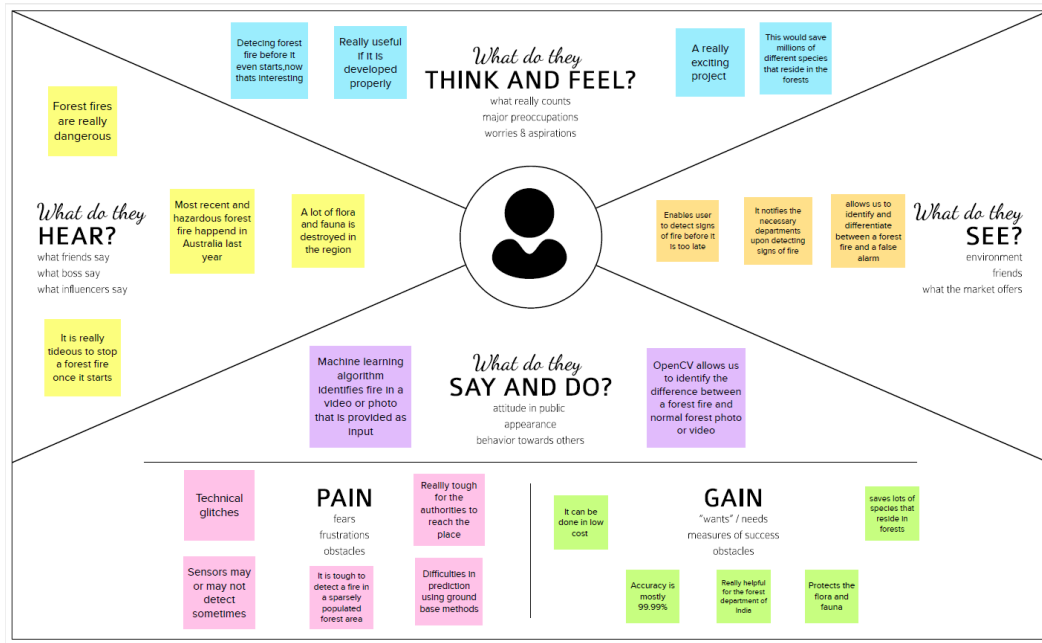
- Jiawei Han, Micheline Kamber, Jian Pei (2012). Data Mining Concepts and Techniques, Third edition, 248-253,350-351.

2.3 Problem Statement Definition

- In the past, fires were detected by 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 whether the fire is burning or not.
- But this approach was slow because the fire may have spread in the large areas and caused a lot of damage before the rescue team arrived.
- Since it is impossible to place a man in every part of a forest, it's important to have monitoring devices in certain areas so we can keep an eye on the forest.
- Both watching towers and satellite images failed to detect the presence of a fire early on, 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

Template



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

➦

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

➦

Warm gathering

Define who should participate in the session and send an invite. Share relevant information as pre-work ahead.

➦

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

➦

Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

Open article

➦

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

response
How might we [your problem statement]?

⌕

Key rules of brainstorming

To run an smooth and productive session

1 Stay in topic	➦ Encourage wild ideas
2 Deferr judgment	➦ Listen to others
3 No too extreme	➦ If possible, be visual



Need some inspiration?

Here's a featured version of this template as featured your work.

Open examples

Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

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

10 minutes

TIP

The idea is to write a sticky note and then use your gut to decide if it's a good idea to stick it (don't overthink!)

Pranav Raj N K

Optical and infrared sensors can be used

CNN should be used to train ML model

Officers should be provided with statistics of most vulnerable areas

Heat map generation using infrared sensors attached to drones

Selvakumar

Influx of regular satellite data

use IIR sensors and lidar sensors for depth tracking

well trained openCV based ML model

using the YOLOv3 software for image processing

Ganesh S

Regular removal of the dry leaves

data collection can be done by flying drones around the forest

Detecting the forest fires by temperature monitoring at regular intervals

Detecting Forest fire by carbon dioxide and black carbon

Pravin Kumar

Detecting forest fire using movement of birds animals

alert system to alert the authorities

Monitoring the forest with drones in regular intervals

Monitoring the carbon trace regularly in most vulnerable areas

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 can break it up into smaller sub-groups.

20 minutes

Detecting fire using temperature monitor

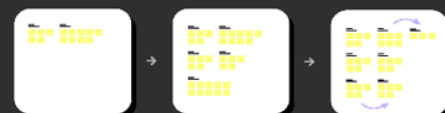
Terrestrial image based 3D modelling

Detecting fire using YOLOv3

Using drones to monitor most vulnerable areas

Infrared cameras for producing heat maps

Alert systems to alert the necessary authorities



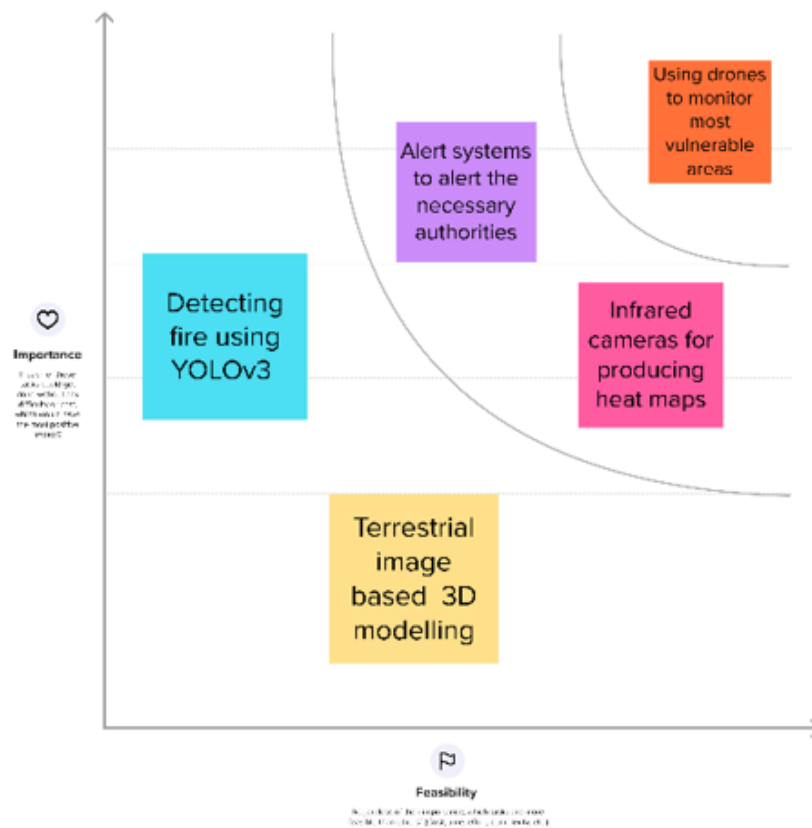
Step-3: Idea Prioritization

4

Prioritize

Your teams 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



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>Statement: To find emerging methods for early detection of forest fires using artificial intelligence.</p> <p>Description: This technology is to be implemented to locate a forest or a bush fire based on the concept of deep learning and YOLO algorithm. After detecting, authorities are to be alerted immediately to mitigate any damage.</p>
2.	Idea / Solution description	<ol style="list-style-type: none"> 1. In case of forest fire detection, the burning substances are primarily identified as sceptical flame regions using a division strategy to expel the non-fire structures and results are verified by deep learning model. 2. The technology used to locate a forest or a bush fire is based on the concept deep learning and YOLO algorithm.
3.	Novelty / Uniqueness	<ol style="list-style-type: none"> 1. Accurate and reliable recognition of sceptical flame regions by means of using YOLO v3 algorithm. 2. Unlike previous algorithms, the exact location of the origin of the forest fire is also detected and sent to the application.
4.	Social Impact / Customer Satisfaction	<ol style="list-style-type: none"> 1. Since we are detecting the outbreak of the fire before it is too big, loses of life, destruction of various environmental, geographical resources can be avoided. 2. Can stop the emission of co2 into the atmosphere and other toxic gases.
5.	Business Model (Revenue Model)	<ol style="list-style-type: none"> 1. The software platform to provide the fully autonomous processing of data received from the camera of UAV to obtain live feed in application. 2. This can also be deployed as a mobile application for easy accessibility.
6.	Scalability of the Solution	<ol style="list-style-type: none"> 1. This application can be developed as a world-wide surveillance system to monitor different forests.

3.4 PROBLEM SOLUTION FIT

Problem-Solution fit canvas 2.0

Purpose: Emerging Methods For Early Detection of Forest Fires/ Team ID:PNT2022TMID23374

Define CS, fit into	1. CUSTOMER SEGMENT(S) <i>Who is your customer?</i>	6. CUSTOMER <i>What constraints prevent your customers from taking action or limit their choices</i>	5. AVAILABLE SOLUTIONS <i>Which solutions are available to the customers when they face the or need to get the job done? What have they tried in the past? What</i>	Explore AS.
	<p>1. Federal agencies(forest fire management) such as National Disaster Management Authority (NDMA) USDA's Forest Service.</p> <p>2.The Department of the Interior's Bureau of Indian Affairs, Bureau of Land Management, Fish and Wildlife Service, and National ParkService.</p>	<p>1. The triple constraint theory says that every project will include three constraints: budget/cost, time, and scope. And these constraints are tied to each other. Any change made to one of the triple constraints will have an effect on the other two.</p> <p>2.With any project, there are limitations and risks that need to be addressed to ensure the project's ultimate success.</p>	<p>From previous studies the available prototype model uses common sensors like Flame sensor ,temperature sensor, gas sensor for fire detection those sensors are attached to trees animals and birds in the forest to detect the forest fire.</p> <p>Pros of existing solutions:</p> <p>1.The forest fire area can be detected and can be located precisely.</p> <p>Cons of existing solutions:</p> <p>1.Complicated to manage.</p> <p>2.Sensor attached to the animals and birds will affect their habitat.</p>	
Focus on J&P, tap into BE.	2. JOBS-TO-BE-DONE / PROBLEMS <i>Which jobs-to-be-done (or problems) do you address for your</i>	9. PROBLEM ROOT CAUSE <i>What is the real reason that this problem exists? What is the back story behind the need</i>	7. BEHAVIOUR <i>What does your customer do to address the problem and get the i.e. directly related: find the right solar panel installer, calculate usage and</i>	Focus on J&P, tap into BE.
	<p>The process provides broad and detailed customer insights that are superior to typical market research methods and critical to developing better solutions for customers. It helped us understand a new space and identify the understand needs so we could enter a new market in a differentiated manner</p>	<p>1. The first step when performing root cause analysis is to analyze the existing situations. This is where the team identifies the factors that impact the problematic event. The outcome of this step is a statement that comprises the specific problem A small team is tasked with the definition of the problem. This could be research staff who assesses and analyzes the situation.</p> <p>2. It describes the difference between the actual conditions and desired conditions.</p>	<p>1. The first step when performing root cause analysis is to analyze the existing situations. This is where the team identifies the factors that impact the problematic event. The outcome of this step is a statement that comprises the specific problem A small team is tasked with the definition of the problem. This could be research staff who assesses and analyzes the situation.</p> <p>2. It describes the difference between the actual conditions and desired conditions.</p>	

Identify strong TR & EM	3. TRIGGERS <i>What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.</i>	10. YOUR SOLUTION <i>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</i>	8.CHANNELS of BEHAVIOUR 8.1.ONLINE <i>What kind of actions do customers take online? Extract online channels from #7</i>	Extract online & offline CH of BE
	<p>Human-caused fires are the result of abandoned campfires unattended, burning debris, equipment use and malfunctions, discarded due to negligence cigarettes and arson</p>	<p>In case of forest fire detection the burning substances are primarily identified as sceptical flame regions using a division strategy to expel the non-fire structures and results are verified by a deep learning model. The technology used to locate a forest or a bush fire is based on the concept of deep learning and YOLO algorithm. This deep learning model is deployed on a UAV which helps in detection of fire, meanwhile it can be monitored by web application and the forest fire area can be located in order to prevent it in advance.</p>	<p>Collect the date and form a dataset in order to compare the flames regions for forest fire detection</p>	
Identify strong TR & EM	4. EMOTIONS: BEFORE / AFTER <i>How do customers feel when they face a problem or a job and afterwards? i. e. lost, insecure > confident, in control - use it in your communication strategy & design.</i>	10. YOUR SOLUTION <i>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</i>	8.CHANNELS of BEHAVIOUR 8.2.OFFLINE <i>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</i>	Extract online & offline CH of BE
	<p>BEFORE: Encroachment through loss of diversity, reduced wildlife</p> <p>AFTER:Forest surveillance systems can be used to monitor stress in the forest so we can prevent human and wildlife and economic damage.</p>	<p>In case of forest fire detection the information is sent to forest authorities so that they will prevent it at ease.</p>	<p>In case of forest fire detection the information is sent to forest authorities so that they will prevent it at ease.</p>	

Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license Created by Daria Nepriakhina / Amaltama.com

AMALTAMA

4.REQUIREMENT ANALYSIS

4.1Functional 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 through remotecontrol
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:

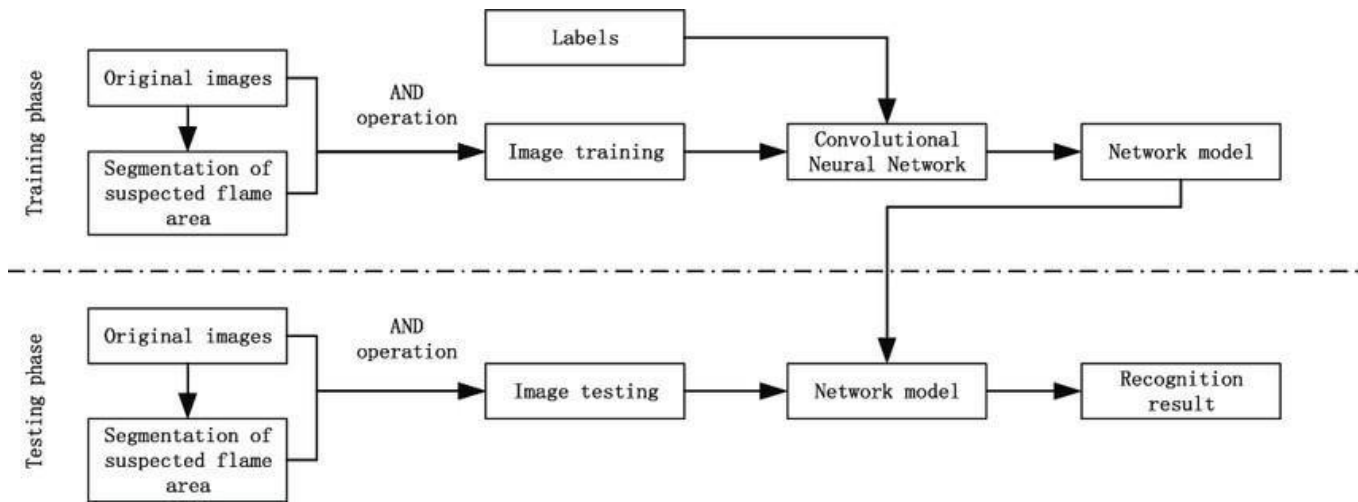
Following are the non-functional requirements of the proposed solution.

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	<i>Model will achieve high accuracy</i>
-------------	--------------------	--

5.PROJECT DESIGN

5.1.Data Flow Diagrams



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:

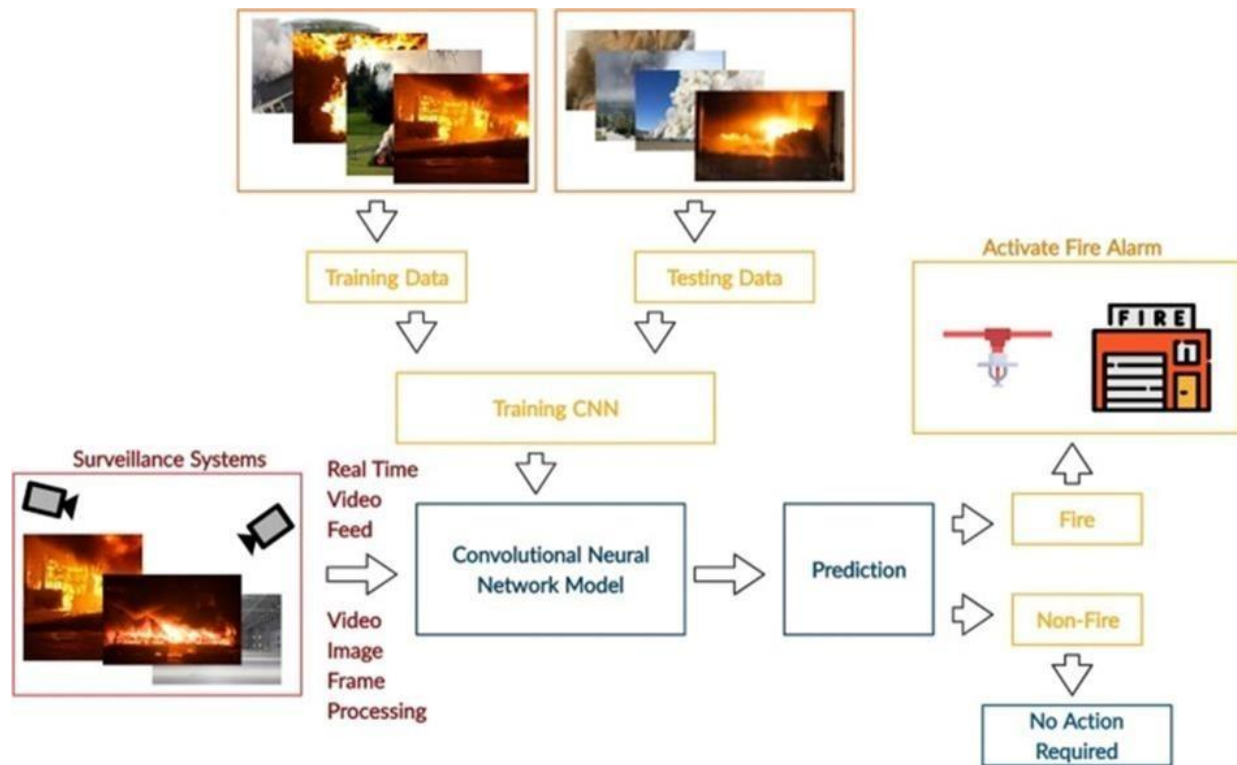
The second part of this step will involve the Rectified Linear Unit or ReLU. We will cover ReLU layers and explore how linearity functions in the context of Convolutional Neural Networks.

Not necessary for understanding CNNs, but there's no harm in a quick lesson to improve your skills.

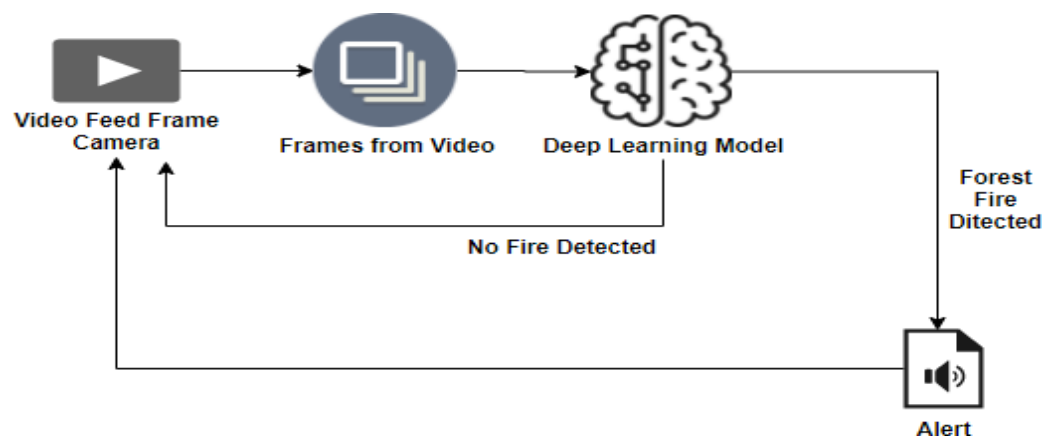
Step 3-Pooling: In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made 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 Neural Networks.

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



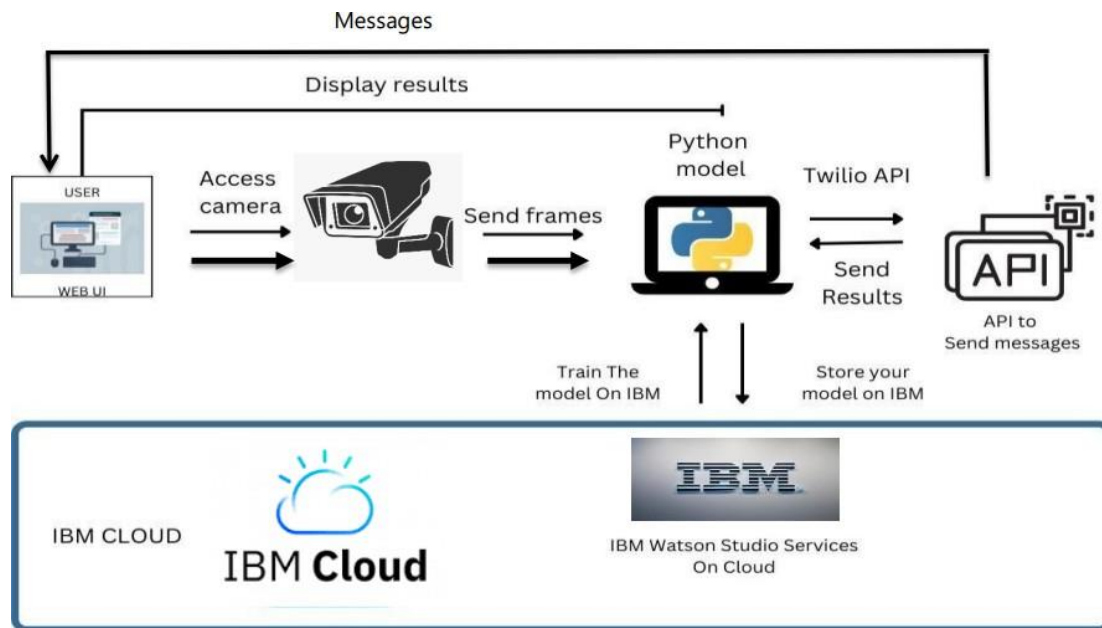


Table 1: Components and Technologies

S.No	Component	Description	Technology
1.	User Interface	The user uses the console to access the interface	Python/HTML , CSS , JavaScript and reactJs
2.	Input	Video Feed	Web Camera/Videon on a site
3.	Conversion	Video inputted is converted into Frames	Frame Converter

4.	Feeding the Model	The Frames are sent to the Deep learning model	Our Model
5.	Dataset	Using Test set and Train set , train the model	Data set from Cloud Storage, Database
6.	Cloud Database	The model is trained in the cloud more precise with	IBM Cloud ant, PythonFlask.

		Detections later images can be added.	
7.	Infrastructure (Server / Cloud), API	Application Deployment on Local System / Cloud Local, Cloud Server Configuration, Twilio API to Send messages	Java/python, React.Js, JavaScript, HTML, CSS ,IBM, Cloud, OpenCV, anaconda navigator

Table 2 : Application Characteristics

s. n o	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 SecurityControl is used	e.g.SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	High scalability with 3-tier architecture	Web server – HTML, CSS, JavaScript Application server, Python, Anaconda Database server – IBM DB2

4.	Availability	Use of load balancing to distributetraffic across servers	IBM load balancer
5.	Performance	Enhance the performance by usingIBM CDN	IBM Content DeliveryNetwork

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
Environmentalism	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	Pranav Raj N K Ganesh S Pravin Kumar R Selvakumar R
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	Pranav Raj N K Ganesh S Pravin Kumar R Selvakumar R
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	Pranav Raj N K Ganesh S Pravin Kumar R Selvakumar R

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	Pranav Raj N K Ganesh S Pravin Kumar R Selvakumar R
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	Pranav Raj N K Ganesh S Pravin Kumar R Selvakumar R

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

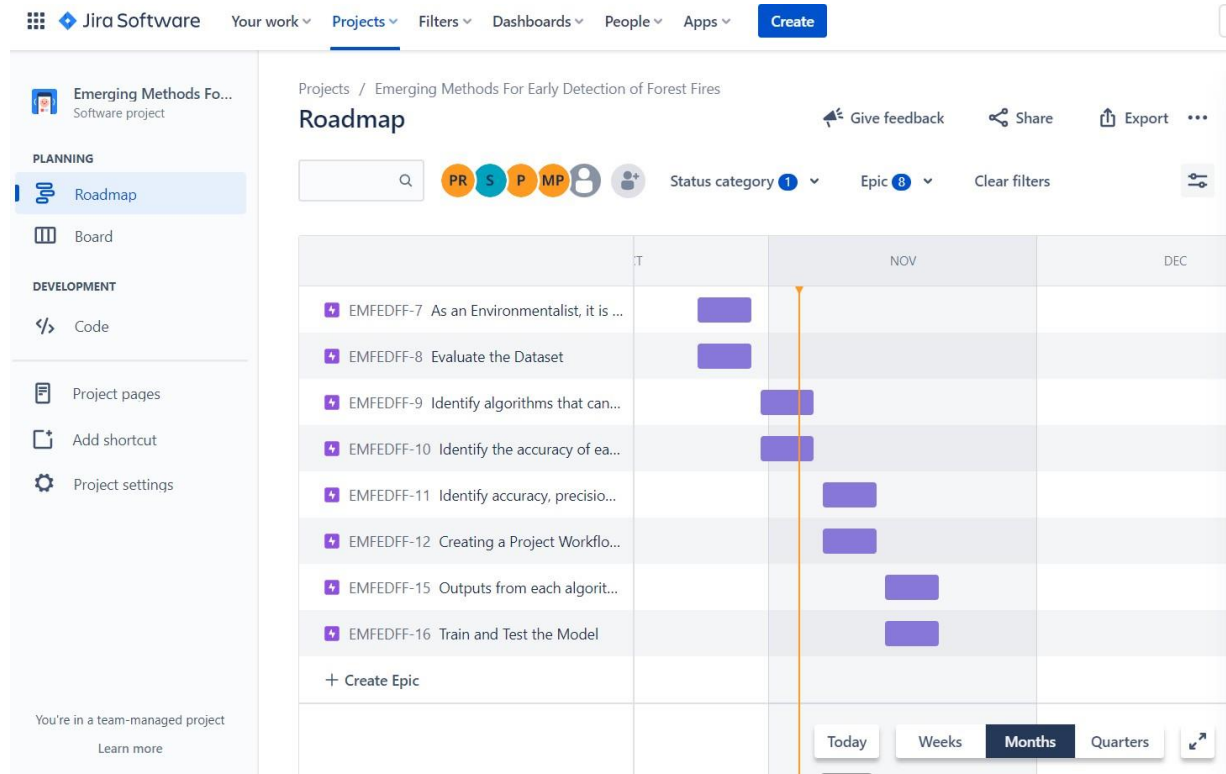
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	7	08 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

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

6.3 Reports from JIRA



7.CODING AND SOLUTION

7.1.Feature 1

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

```
[ ] 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).

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

Image Pre-processing

```
1 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.

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

```
2 #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		

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

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

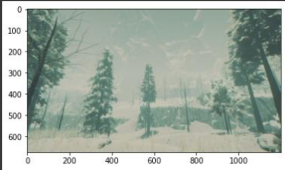
Epoch 1/15
4/14 [====>.....] - ETA: 16s - loss: 2.6860 - accuracy: 0.7521 - mse: 0.2385WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least 'steps'
14/14 [-----] - 15s 929ms/step - loss: 2.6860 - accuracy: 0.7521 - mse: 0.2385 - val_loss: 1.1237 - val_accuracy: 0.7969 - val_mse: 0.1896


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

[ ] #Testing the model
from keras.models import load_model
import cv2
from matplotlib import pyplot as plt
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"))


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

1/1 [=====] - 0s 83ms/step
NO FOREST FIRE DETECTED
```




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

1/1 [=====] - 0s 127ms/step
FOREST FIRE DETECTED! SMS SENT!
```



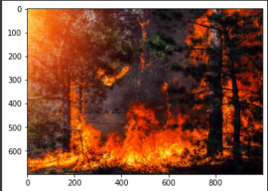
```
[ ] prediction(r'/content/Dataset/Dataset/test_set/forest/111188170_river_in_the_mountain_forest.jpg')

1/1 [=====] - 0s 51ms/step
NO FOREST FIRE DETECTED
```



```
[ ] prediction(r'/content/drive/MyDrive/Dataset/train_set/with fire/with fire (102).jpg')

1/1 [=====] - 0s 50ms/step
FOREST FIRE DETECTED! SMS SENT!
```



7.2.Feature 2

Model deployment in IBM

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

ffd_model.h5

[ ] !pip install watson-machine-learning-client

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
    |#####| 538 kB 5.1 MB/s
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2022.9.24)
Collecting ibm-cos-sdk
  Downloading ibm-cos-sdk-2.12.0.tar.gz (55 kB)
    |#####| 55 kB 3.8 MB/s
Collecting boto3
  Downloading boto3-1.26.16-py3-none-any.whl (132 kB)
    |#####| 132 kB 45.5 MB/s
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (0.8.10)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (4.64.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.3.5)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2.23.0)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.24.3)
Collecting lomond
  Downloading lomond-0.3.3-py2.py3-none-any.whl (35 kB)
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
    |#####| 79 kB 7.2 MB/s
Collecting botocore<1.30.0,>=1.29.16
  Downloading botocore-1.29.16-py3-none-any.whl (9.9 MB)
    |#####| 9.9 MB 44.2 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting urllib3
  Downloading urllib3-1.26.13-py2.py3-none-any.whl (140 kB)
    |#####| 140 kB 64.3 MB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from botocore<1.30.0,>=1.29.16->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.30.0,>=1.29.16->boto3->watson-machine-learning-client) (1.15.0)
Collecting ibm-cos-sdk-core==2.12.0
```

```
[ ] !pip install ibm_watson_machine_learning

Downloading ibm-cos-sdk-2.7.0.tar.gz (51 kB)
    |#####| 51 kB 718 kB/s
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2.28.1)
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: urllib3 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.26.13)
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: tabulate in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.8.10)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (4.13.0)
Collecting ibm-cos-sdk-core==2.7.0
  Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)
    |#####| 824 kB 43.3 MB/s
Collecting ibm-cos-sdk-s3transfer==2.7.0
  Downloading ibm-cos-sdk-s3transfer-2.7.0.tar.gz (133 kB)
    |#####| 133 kB 54.2 MB/s
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.0->ibm_watson_machine_learning) (0.10.0)
Collecting docutils<0.16,>=0.10
  Downloading docutils-0.15.2-py3-none-any.whl (547 kB)
    |#####| 547 kB 46.1 MB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.0->ibm_watson_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2022.6)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.21.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.7.0->ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.10)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (4.1.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (3.10.0)
Requirement already satisfied: pyparsing<3.0.5,>=2.8.2 in /usr/local/lib/python3.7/dist-packages (from packaging->ibm_watson_machine_learning) (3.0.9)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer
  Building wheel for ibm-cos-sdk (setup.py) ... done
  Created wheel for ibm-cos-sdk: filename=ibm-cos_sdk-2.7.0-py2.py3-none-any.whl size=72567 sha256=7832c5956ec8c831f53329b13e8107d7c1ed585fb4a3715b9077f3f21d21de41
  Stored in directory: /root/.cache/pip/wheels/47/22/bf/e154ff0f5de93c477ac0ca69abfb8b799c5b28a66b44c2
  Building wheel for ibm-cos-sdk-core (setup.py) ... done
  Created wheel for ibm-cos-sdk-core: filename=ibm-cos_sdk_core-2.7.0-py2.py3-none-any.whl size=581011 sha256=5a368e4936387279eefbcbfce72b03b3c55cf7210d3719f20ff4d15eb6254
  Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17cfd02c1369281f3d22aaf5902c19
  Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done
  Created wheel for ibm-cos-sdk-s3transfer: filename=ibm-cos_sdk_s3transfer-2.7.0-py2.py3-none-any.whl size=88618 sha256=532075cbdc7829538d8ea3c2898233aa0408c25a1c2e05fd6ee9d0e434a7e1bc
  Stored in directory: /root/.cache/pip/wheels/5f/b7/14/fbe02b2c1ef1af980650c7e51743d1c83890852e598d164b9da
Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer
Installing collected packages: docutils, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer, ibm-cos-sdk, ibm-watson-machine-learning
Attempting uninstall: docutils
```

```
#Connecting to IBM Cloud from Notebook
from ibm_watson_machine_learning import APIClient
credentials = {
    'url': 'https://us-south.ml.cloud.ibm.com',
    'apikey': 'NPAIYP4IHrGISRE4H1b_HFX3RwVfgB1VwJBx4040E30F'
}
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.

Client

<ibm_watson_machine_learning.client.APIClient at 0x7f1f78c5c7d0>

Client.spaces.get_details()

{'resources': [{'entity': {'compute': {'crn': 'crn:v1:bluemix:public:pm-28:us-south:a/5d5fb75a19354d0cbaf821dfe7c9a83e:3de9f192-19f2-4fa4-8ace-20b1e12f332::',
    'guid': '3de9f192-19f2-4fa4-8ace-20b1e12f332',
    'name': 'Watson Machine Learning-wp',
    'type': 'machine_learning'}},
    'description': '',
    'name': 'Forest fire project',
    'scope': {'bss_account_id': '5d5fb75a19354d0cbaf821dfe7c9a83e'},
    'stage': {'production': False},
    'status': {'state': 'active'},
    'storage': {'properties': {'bucket_name': 'f0719962-137d-48ed-8261-7c178fe1abe3',
    'bucket_region': 'us-south',
    'credentials': {'admin': {'access_key_id': 'Acc6552a1c4243549d3dfa7384b17611',
    'api_key': 'uIv1w55stFmVt02mclt5BfGdyru7LNe1TFykXak9A7',
    'secret_access_key': '1ce88c12428abca1ed4bef280c57c6eda423f2e5244edf00e',
    'service_id': 'ServiceId:99a89513-a4a0-4869-3760-38c7bac30021'},
    'editor': {'access_key_id': '1148113fc59546ce993d141680fdd171',
    'api_key': 'mewZhoxs-3xeSk09wqIiLqNeb_pVf47q8Z5xFU195V',
    'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/5d5fb75a19354d0cbaf821dfe7c9a83e:976296e3-4d56-45bd-bcf2-1a6b7a147a5f::',
    'secret_access_key': '45e8db65ed1871a4d339bf44468a5978659db9c8bedeedic',
    'service_id': 'ServiceId:c4be00b5-bfd9-4a0a-8456-25d54140a187'},
    'viewer': {'access_key_id': 'd17d86f68ed34e998218215155885ce2',
    'api_key': '459y5Jlym1mBqHqk4KpiVhSVIM2RdZJLa56Peys72',
    'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/5d5fb75a19354d0cbaf821dfe7c9a83e:976296e3-4d56-45bd-bcf2-1a6b7a147a5f::',
    'secret_access_key': '5dc0aefc3a35e0482ce55d18ed5c8519b3c7d36e9ef8222',
    'service_id': 'ServiceId:d73d5e5-1c5d-4588-9371-76e1143a38c6'}},
    'endpoint_url': 'https://s3.us-south.cloud-object-storage.appdomain.cloud',
    'guid': '976296e3-4d56-45bd-bcf2-1a6b7a147a5f',
    'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/5d5fb75a19354d0cbaf821dfe7c9a83e:976296e3-4d56-45bd-bcf2-1a6b7a147a5f::',
    'type': 'bmcos_object_storage'}},
    'metadata': {'created_at': '2022-11-12T09:18:31.403Z',
```

```
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
6a0d94f6-58b3-4c8f-a11c-cab2512b6969 Deployment_model 2022-11-10T15:53:24.164Z
c3920229-e32f-47da-aa5b-fac79d47f3f4 Forest fire project 2022-11-12T09:18:31.403Z

space_id = 'c3920229-e32f-47da-aa5b-fac79d47f3f4' #Space User ID
space_id

'c3920229-e32f-47da-aa5b-fac79d47f3f4'

#Setting created deployment space as default
Client.set_default_space(space_id)

'SUCCESS'
```

```
#Seeing tensorflow asset id
Client.software_specifications.list()

NAME ASSET_ID TYPE
default_py3.6 0603a0c8-8b7d-44a0-a909-46c416adcb9 base
kernel-spark3.2-scala2.12 020d69ce-7ac1-5e08-ac1a-31189807356a base
pytorch-onnx 1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base
scikit-learn 0.20-py3.6 09c5a1d0-9c1e-4473-a344-e7b665ff687 base
spark-mllib 3.0-scala 2.12 09facff0-90a7-5899-b9ed-1ef348aebdee base
pytorch-onnx rt22.1-py3.9 0b848dd4-e681-5599-be41-b5fefcc6471 base
ai-function 0.1-py3.6 0cd0bf1e-5376-4f4d-92dd-dab369a99bda base
shiny-r3.6 0ee679df-875e-4f24-8ae9-62dc2148306 base
tensorflow 2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base
pytorch 1.1-py3.6 10ac12d6-b030-4ccd-8392-3e922c096a92 base
tensorflow 1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base
autoai-kb_rt22.2-py3.10 125b6d9a-5b1f-5e80-972a-b251688ccf40 base
runtime 22.1-py3.9 12b03a17-24d8-5082-90ef-6ab31f7fd3eb base
scikit-learn 0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base
default_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base
pytorch-onnx 1.3-py3.6 1bc6029a-c97-56da-b8e0-39c3880dbbe7 base
kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cd5f598 base
pytorch-onnx rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base
tensorflow 2.1-py3.6 1eb23ba8-d6ed-5dfe-bda5-3f0af1665666 base
spark-mllib 3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base
tensorflow 2.4-py3.8-horovod 217c16fe-178f-56bf-824a-b19f20564c49 base
runtime 22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce58 base
do_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base
autoai-ts 3.8-py3.8 2a0c932-798f-5ae9-ab06-15e0c2482fb5 base
tensorflow 1.15-py3.6 2b73a275-7cbf-420b-e212-eae7f43ae0bc base
kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a401-482c8368839a base
pytorch 1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base
spark-mllib 2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base
pytorch-onnx 1.1-py3.6-edt 32983cea-3f32-4408-8965-dde874a8d67e base
spark-mllib 3.0-py37 3650e8be-8770-55ba-ab2a-eafe787000e9 base
spark-mllib 2.4 306ad210-65b0-4fac-9d55-d7ed0a2c1326 base
autoai-ts_rt22.2-py3.10 396b2e83-0953-5b06-9a55-7ce1628a406f base
xgboost 0.82-py3.6 39e31acd-5f30-41dc-a444-60233c88306e base
pytorch-onnx 1.2-py3.6-edt 40589d0e-7819-4e28-8daa-fb03b6f4fe12 base
pytorch-onnx rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b94291431 base
default_r3py38 41c247d3-45f8-5a71-b065-8580229facf0 base
autoai-ts_rt22.1-py3.9 4269d25e-07b0-5d40-8f66-2d093b0c7177 base
autoai-ohm 3.0 42b92e18-d9ab-567f-989a-4240a1ad5d57 base
pmml 3.0 4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base
spark-mllib 2.4-r 3.6 49483dff-92e9-4c87-a3d7-a42d0021c095 base
xgboost 0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base
pytorch-onnx 1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base
```

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

'acd9c798-6974-5d2f-a657-ce06e986df4d'

[ ] 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
})

model_details

[ ] {'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': '2022-11-24T05:10:31.424Z',
    'id': 'e28bd328-37ba-47d9-9dfd-8e1b0c39d888',
    'modified_at': '2022-11-24T05:10:51.515Z',
    'name': 'forest fires project',
    'owner': 'IBMId-668800EGUX',
    'resource_key': '44bbf2f7-7a2c-4754-9dd3-e0443b83882e',
    'space_id': 'c3020229-e32f-47da-aa5b-fac79d47f3f4',
    'system': {'warnings': []}}

[ ] model_id = Client.repository.get_model_uid(model_details)
model_id

This method is deprecated, please use get_model_id()
'e28bd328-37ba-47d9-9dfd-8e1b0c39d888'

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

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

```
[ ] !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 (7.15.3)
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: 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) (2.28.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.26.13)
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)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)

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 (7.15.3)
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) (2.28.1)
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: 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) (1.26.13)
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: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)

[ ] 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)
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=7038 sha256=6ea51418eaf15c3a76b6b68d2e4f525764d4a3de9e313da7b0fc64aa746e33c73
  Stored in directory: /root/.cache/pip/wheels/ba/fb/ea/57c0146b66d4ca3a0ada4199b0ecb5f9dfcb7b7e22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
```

```
[ ] #Import opencv library
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
#Import 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.

#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']

[ ] account_sid = 'AC87c017e7422895ac0fdd7de1bd59466'
auth_token = 'a5281ae2a4149b2a0721e35b07e6d89b'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
        body='Forest fire is detected , stay alert!',
        from_='+12802721418',
        to='+910382118046'
    )

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

59B4531C8c26b16916c76378541c2b85db
Fire Detected
SMS Sent
```

8.TESTING

8.1.Test Cases

TEST CASES REPORT

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		Pranav Raj N K Ganesh S
OP_RT_002	Functional	Page	Check if user cannot upload unsupported files	1) The sensor senses the fire 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	BUC_HP_002	Selvakumar R Pravin Kumar R
OP_RT_003	Functional	Page	Checks whether the page redirects to the result page to the given output	1) The sensor senses the fire 2) checks with the pre-uploaded images 3) checks if there is fire detection	Sample 1.png	The page should redirect to the results page	Working as expected	PASS		Pranav Raj N K
MB_RT_001	Functional	Backend	Checks if all the routes are working properly	1) The sensor senses the fire 2) checks with the pre-uploaded images 3) checks if there is fire detection	Sample 1.png	All the routes should properly work	Working as expected	PASS		Ganesh S
N_DC_001	Functional	Model	Checks whether the model can handle various sizes	1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input	Sample 1.png Sample 1.XS.png Sample 1.XL.png	The model should rescale the image and predict the results	Working as expected	PASS		Selvakumar R
N_DC_002	Functional	Model	Check if the model predicts the digit	1) Open the page 2) Select the input images	Sample 1.png	The model should predict the number	Working as expected	PASS		Pranav Raj N K Ganesh S
N_DC_003	Functional	Model	Check if the model can handle complex input image	1) Open the page 2) Select the input images 3) Check the results	Complex Sample.png	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL	BUC_M_001	Selvakumar R Pravin Kumar R

RL_DC_001	Functional	Result Page	Verify the elements	1) Open the page 2) Select the input image 3) Check if all the UI elements are displayed properly	Sample 1.png	The Result page must be displayed properly	Working as expected	PASS		Pranav Raj N K Ganesh S
RL_DC_002	Functional	Result Page	Check if that image is displayed properly	1) Open the page 2) Select the input image 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	BUC_RP_001	Selvakumar R Pravin Kumar R
RL_DC_003	Functional	Result Page	Checks whether the displayed prediction is accurate	1) Open the page 2) Select the input image 3) Check if all the other predictions are displayed	Sample 1.png	The other predictions should be displayed properly	Working as expected	PASS		Pranav Raj N K Selvakumar R

9. 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. This 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

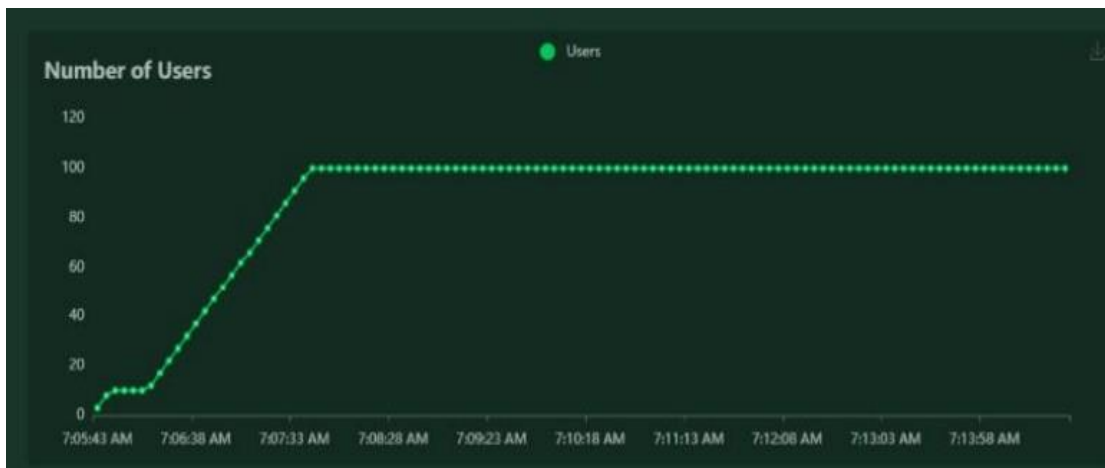
This report shows the number of test cases that have passed, failed, and untested

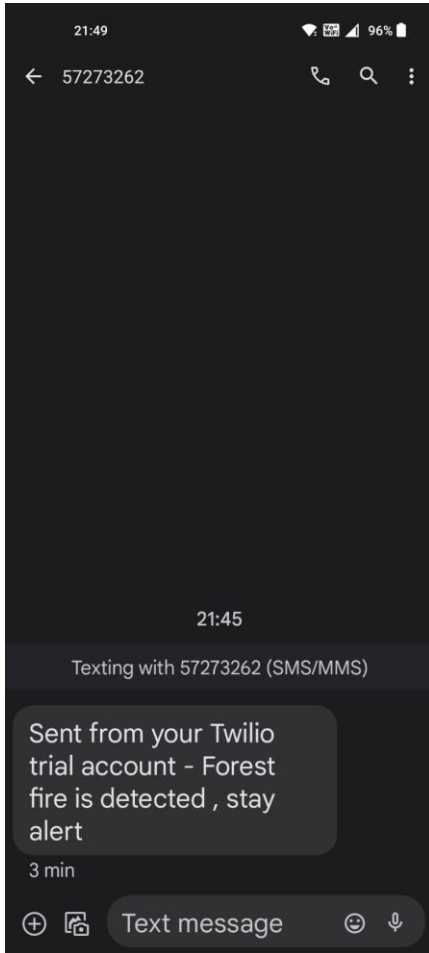
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	0	10
Security	2	0	0	2
Performance	2	0	0	2
Exception Reporting	2	0	0	2
Final Report Output	3	0	0	3

10.Results

10.1.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





11. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. The proposed model can be used in combination with a night camera 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 accurate outcome 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 because to 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 forest more 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.

12.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.

13.FUTURE SCOPES

The 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.

14.APPENDIX

14.1.Source Code

```
[ ] | 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 (7.15.3)
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: 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) (2.28.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.26.13)
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)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)

❶ | pip install twilio

❷ | 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 (7.15.3)
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) (2.28.1)
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: 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) (1.26.13)
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: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)

[ ] | 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)
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=7038 sha256=6ea5140eaf15e3a76b668d2e4f525764d4a3de9e313da7b0fc64aa746e33c73
  Stored in directory: /root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0ecb5f9dfcb7b7e22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
```

```
[ ] #import opencv library
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
#import 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.

❶ | #load the saved model
model = load_model(r'./content/ffd_model.h5')
#define video
video = cv2.VideoCapture('./content/demo.mpd')
#define the features
name = ['forest','with forest']

[ ] account_sid = 'AC87c017e7422895ac0fd7de1bd4d59466'
auth_token = 'a5281ae2a4149b2a0721e35b07e0d89b'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
        body="forest fire is detected , stay alert",
        from_="+12027212410",
        to="+916382110646"
    )

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

59045318c20b16916c76378541c2b85db
Fire Detected
SMS Sent
```

14.2.GitHub & Project Demo Link

GitHub Link:

<https://github.com/IBM-EPBL/IBM-Project-25046-1659953009>

Project Demo Link:

[https://drive.google.com/file/d/1k13_PQ4s_-TC7cuYCB_zsKbc8Mt2r37-/view?usp=share link](https://drive.google.com/file/d/1k13_PQ4s_-TC7cuYCB_zsKbc8Mt2r37-/view?usp=share_link)

