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 sucha 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 intheir 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 05/1731-35Volume 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

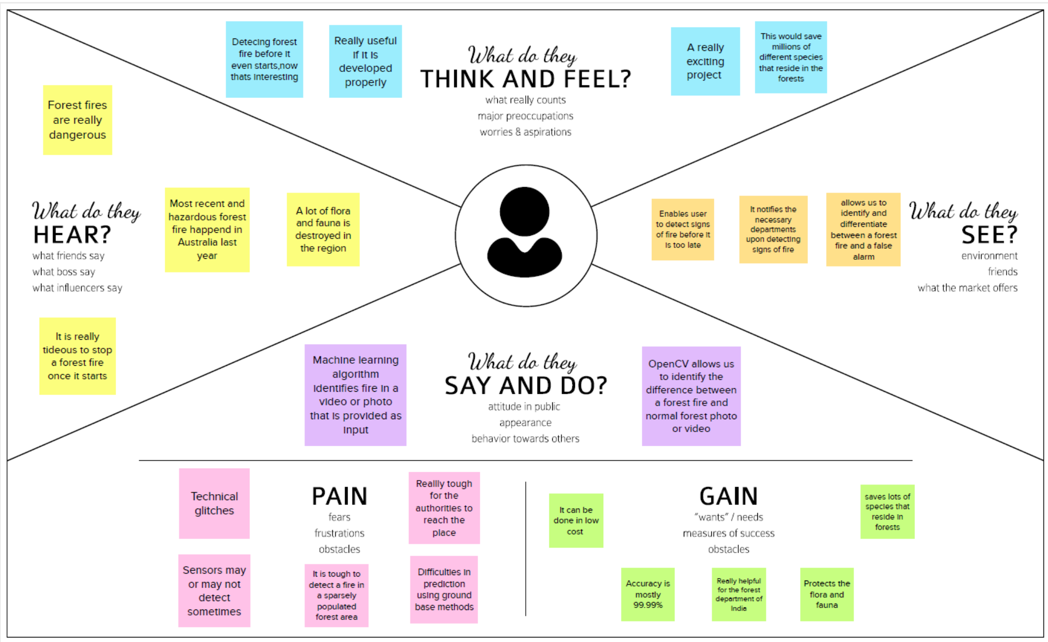
- In the past, fires were detected by watching towers or using satellite images.
- Satellites collect images of fires and send them to a monitoring authority for review. If the images appear to show a fire, the authority will determine whether the fire is burning or not.
- But this approach was slow 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 eyeon 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.

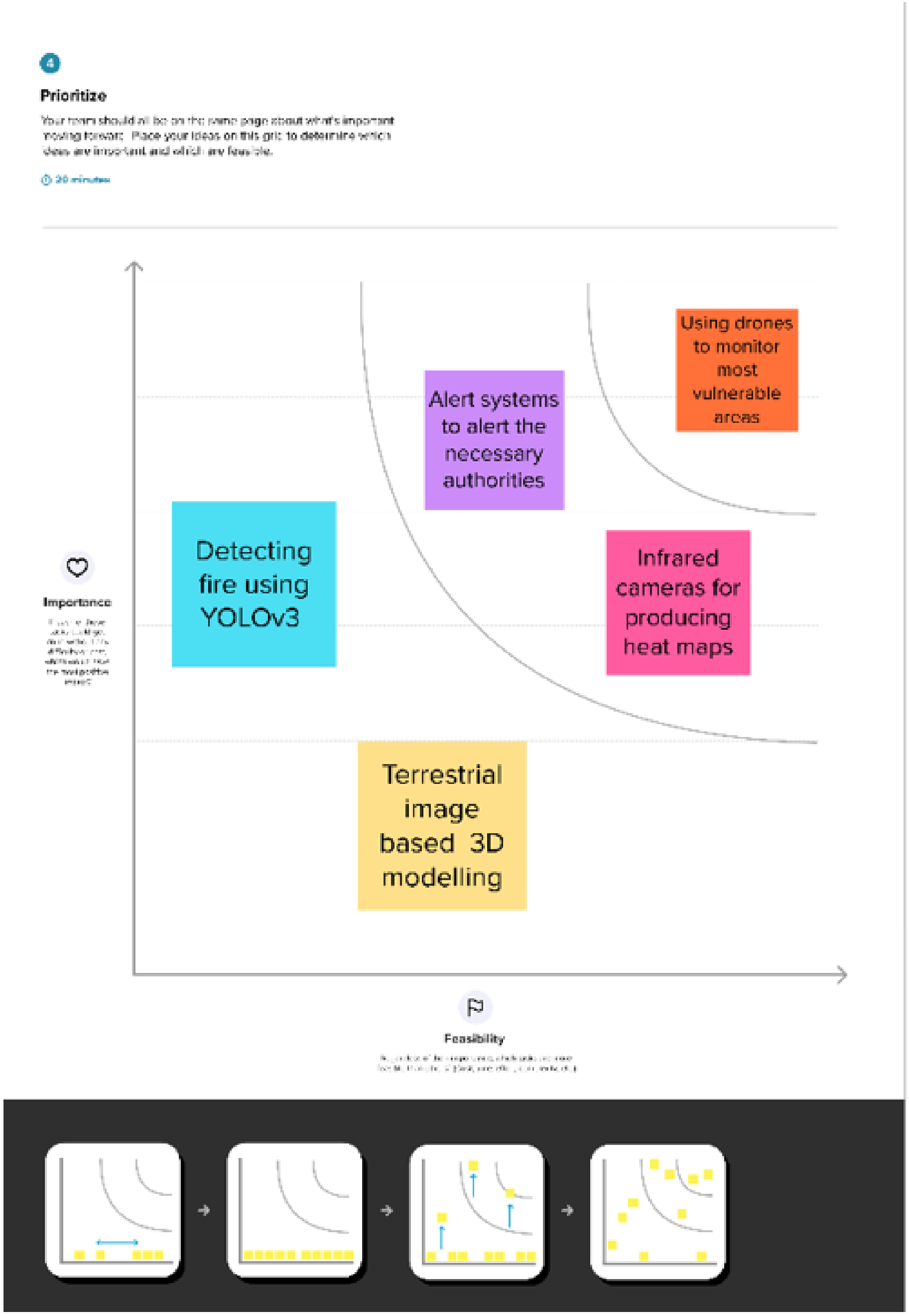
3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement**: To find emerging methods for early detection of forest fires using artificial intelligence. **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.
2.	Idea / Solution description	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	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	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)	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	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) — CS

1. Federal agencies(forest fire management) such as National Disaster Management Authority (NDMA) USDA's Forest Service.

2.The Department of the Interior's Bureau of Indian Affairs, Bureau of Land Management, Fish and Wildlife Service, and National ParkService.

6. CUSTOMER — CC

What constraints prevent your customers from taking action or limit their choices

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.

2.With any project, there are limitations and risks that need to be addressed to ensure the project's ultimate success.

5. AVAILABLE SOLUTIONS — AS

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

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.

Pros of existing solutions:

1.The forest fire area can be detected and can be located precisely.

Cons of existing solutions:

1.Complicated to manage.

2.Sensor attached to the animals and birds will affect their habitat.

Explore AS,

Focus on J&P, tap into BE,

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

Which jobs-to-be-done (or problems) do you address for your

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

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists? What is the back story behind the need

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.

2. It describes the difference between the actual conditions and desired conditions.

7. BEHAVIOUR — BE

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

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.

2. It describes the difference between the actual conditions and desired conditions.

Focus on J&P, tap into BE,

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Human-caused fires are the result of abandoned campfires unattended, burning debris, equipment use and malfunctions, discarded due to negligence cigarettes and arson

4. EMOTIONS: BEFORE / AFTER — EM

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.

BEFORE: Encroachment through loss of diversity, reduced wildlife

AFTER: Forest surveillance systems can be used to monitor stress in the forest so we can prevent human and wildlife and economic damage.

10. YOUR SOLUTION — SL

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.

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.

8. CHANNELS of BEHAVIOUR — CH

8.1ONLINE

What kind of actions do customers take online? Extract online channels from #7

Collect the date and form a dataset in order to compare the flames regions for forest fire detection

8.2OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

In case of forest fire detection the information is sent to forest authorities so that they will prevent it at ease.

Extract online & offline CH of BE

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:

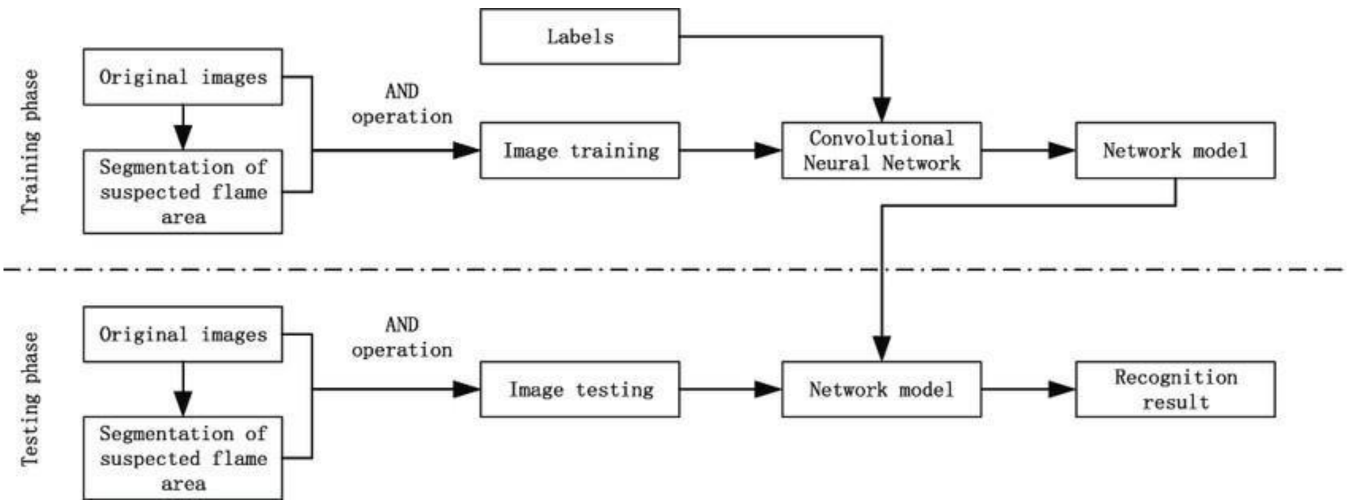
Following are the non-functional requirements of the proposed solution.

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***	***Model will achieve high accuracy***

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:

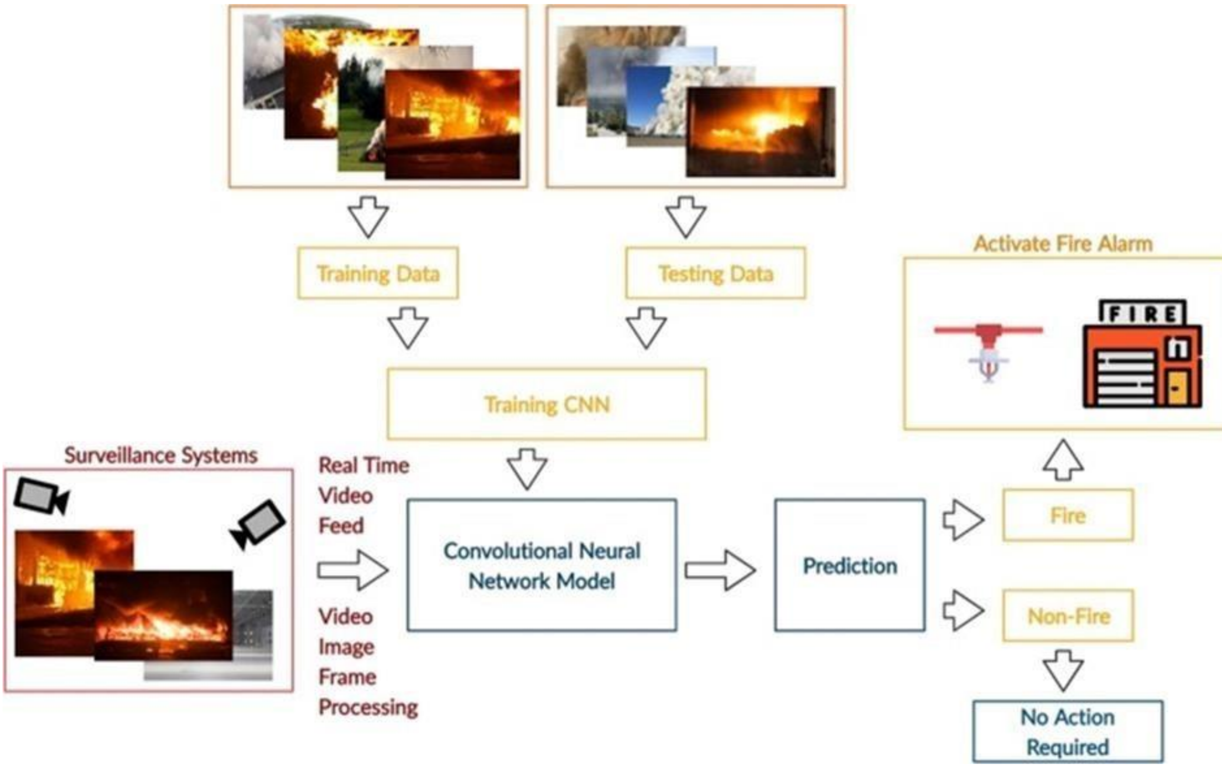
The second part of this step will involve the Rectified Linear Unit or ReLU. We will cover ReLU layers and explore how linearity functions in the context of Convolutional Neural Networks.

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

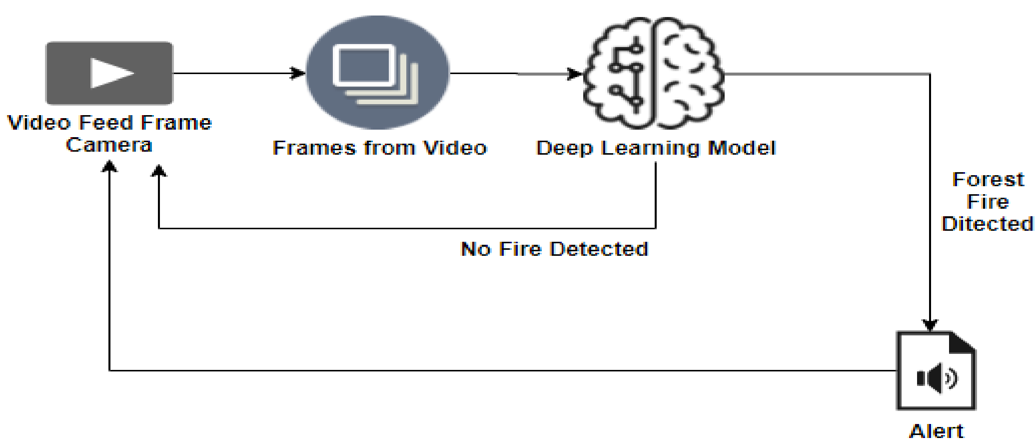
Step 3-Pooling: In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made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



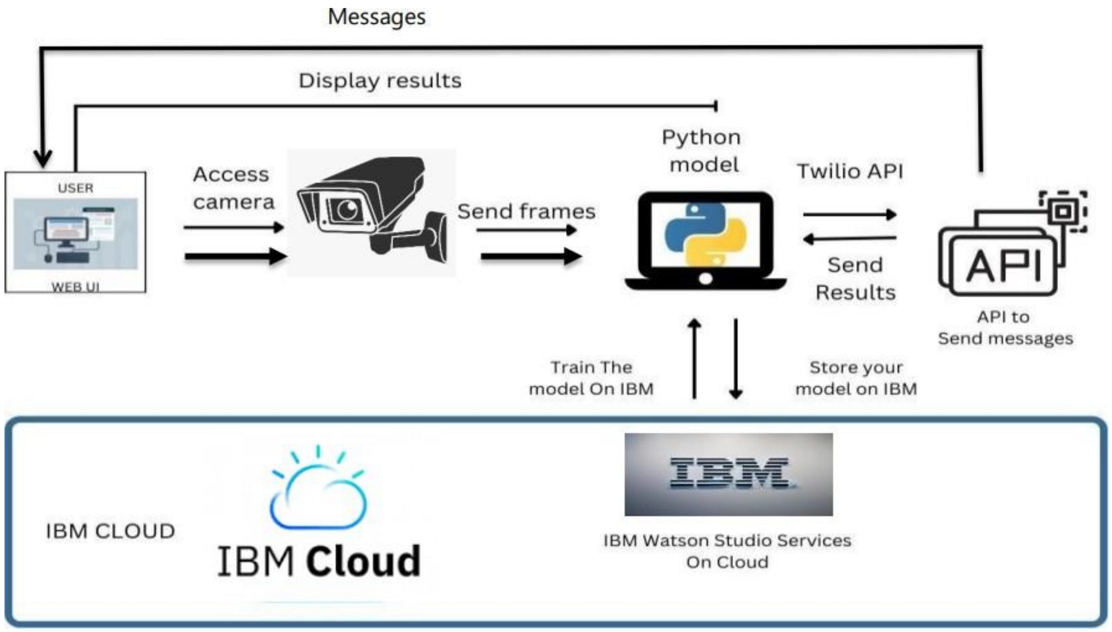


Table 1: Components and Technologies

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

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

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

```
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')
```

```
#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)
```

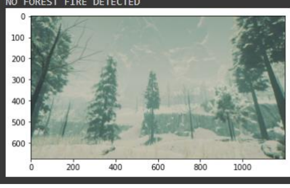
```
Epoch 1/15
 4/14 [=======>.....................] - ETA: 16s - loss: 2.6860 - accuracy: 0.7521 - mse: 0.2385WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `step:
14/14 [==============================] - 15s 929ms/step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

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

```python
#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"))
```

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

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

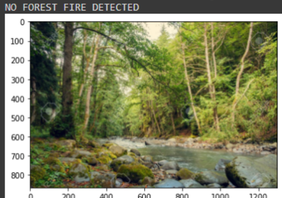



```python
prediction(r'/content/Dataset/Dataset/test_set/with fire/Fire_2_696x392.jpg')
```

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


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

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




```python
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.*->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.*->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-cos-sdk==2.7.*->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,>=2.0.2 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/e1154ff0f5de93cc477acd0ca69abfbb8b799c5b28a66b44c2
      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=501011 sha256=5e368e493638727296efbbc6bfce72b03b3c55cf7210d3719f20ff4d15eb6254
      Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17cfd02c13369281f3d232aaf5902c19
      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/fbe02bc1ef1af890650c7e51743d1c83890852e598d164b9da
    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_HFX3RwYfgB1YwJBx4O4OE3Df"
}
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
```

```
&lt;ibm_watson_machine_learning.client.APIClient at 0x7f1f78c5c7d0&gt;
```

```
Client.spaces.get_details()
```


```
{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:us-south:a/5d5fb75a19354d0cbaf821dfe7c9a83e:3de9f192-19f2-4fa4-8ace-20b11e12f332::',
      'guid': '3de9f192-19f2-4fa4-8ace-20b11e12f332',
      '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': '4cc6552a1c4243549d3dfa7384b17611',
        'api_key': 'uIvIwS5sLFmwVT02mcltSBfGDyru7LNriTFyKdXak9A7',
        'secret_access_key': '1ce86c12420abca1e4bef280c57c6eda423f2e5244edf00e',
        'service_id': 'ServiceId-99e89513-a4a9-4869-a7b0-30c7bac20821'},
       'editor': {'access_key_id': '1148113fc59546ce903d141680fdd171',
        'api_key': 'mwwZhoxs-jxeSkO9wMqIirLqNeb_pVFq7q0Z5xFUi95V',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/5d5fb75a19354d0cbaf821dfe7c9a83e:976296e3-4d56-45bd-bcf2-1a6b7a147a5f::',
        'secret_access_key': '45ed8b65ed1071a4d339bf44468a5b97065bd9c9bedeed1c',
        'service_id': 'ServiceId-c4be00b5-bfd9-4a0a-8456-25d54140a187'},
       'viewer': {'access_key_id': 'd17d86f68ed34e998218215155885ce2',
        'api_key': '459y5jUymGlmBMqHx4kLpivHSViW2RdZJLaS6PpEys72',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/5d5fb75a19354d0cbaf821dfe7c9a83e:976296e3-4d56-45bd-bcf2-1a6b7a147a5f::',
        'secret_access_key': '5dc0afcfc0a35e9d82ce55d18ed5c8519b3c7d36efef8222',
        'service_id': 'ServiceId-d73d5ec5-1c5d-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
6a9d04f6-5883-4c8f-a11c-cab2512b6969  Deployment_model    2022-11-19T15:53:24.164Z
c3920229-e32f-47da-aa5b-fac79d47f3f4  Forest fire project 2022-11-12T09:18:31.403Z
------------------------------------  ------------------  ------------------------
```

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

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

```
#Setting created deployment space as default
Client.set.default_space(space_uid)
```

```
'SUCCESS'
```

```
#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
scikit-learn_0.20-py3.6        09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12     09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9      0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6          0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                     0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod   1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6              10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl      111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10        125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9             12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6        154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                   1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6         1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6           1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt  1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6           1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod   217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda        26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                       295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8            2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6          2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9          2b7961e2-e3b1-5a8c-a491-482c8368839a  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-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37           36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10        396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6             39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt     40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10     40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9         4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                 42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                   493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6          49403dff-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': 'e20bd328-37ba-47d9-9dfd-0e1b0c39d888',
  'modified_at': '2022-11-24T05:10:51.515Z',
  'name': 'forest fires project',
  'owner': 'IBMid-668000EGUX',
  'resource_key': '44bbf2f7-7a2c-4754-9dd3-e0443b83082e',
  'space_id': 'c3920229-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()
'e20bd328-37ba-47d9-9dfd-0e1b0c39d888'
```

```
#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=6ea51410eaf15e3a76b668d2e4f525764d4a3de9e313da7b0fc64aa746e33c73
  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 librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
```

```
WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.
```

```
#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 = 'AC87c017e7422895ac0fdd7de1b4d59466'
auth_token = 'a5281ae2a4149b2a0721e35b07e6d09b'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
      body='Forest fire is detected , stay alert',
      from_='+12202723418',
      to='+916382110646'
     )

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

```
SM04531c8c26b16916c76378541c2b85db
Fire Detected
SMS Sent
```

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. 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

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

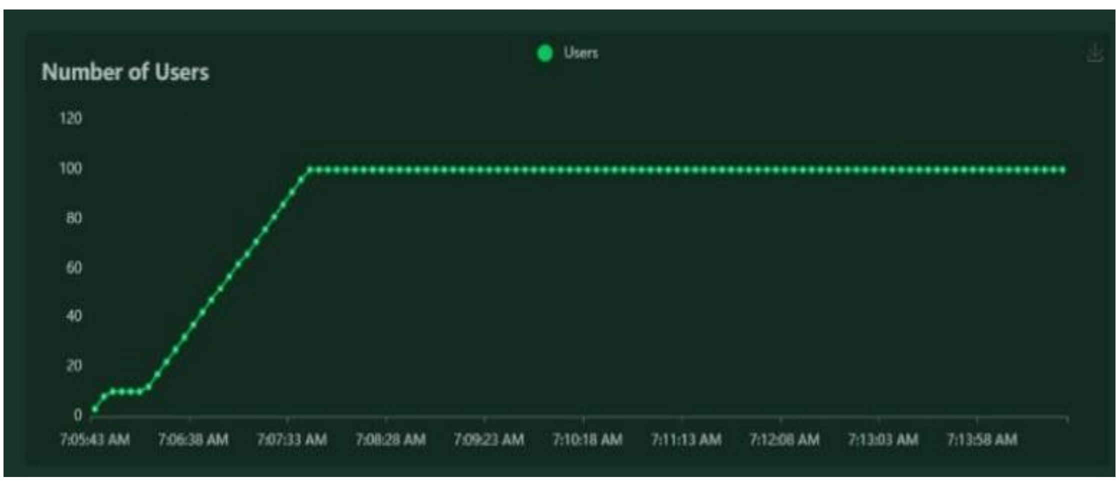
Request Statistics

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**


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


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

11. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. The proposed model can be used in combinationwith a night camera and a thermal camera in a forest to identify tiny fire signs.
2. More datasets and images can be used to trainfor 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 alsocause to trigger the alarm.

APPLICATIONS:

1. It will contribute to surveillance technology thatimproves the accuracy and predictability of fire detection.
2. able to detect the fire forest more precisely, as well assome forest plants and wildlife.
3. Detect the amount of dangers that should be treated and those that should not. extraassistancein 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)

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=6ea51410eaf15e3a76b668d2e4f525764d4a3de9e313da7b0fc64aa746e33c73
  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 librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound

WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.

#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 = 'AC87c017e7422895ac0fdd7de1b4d59466'
auth_token = 'a5281ae2a4149b2a0721e35b07e6d09b'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
      body='Forest fire is detected , stay alert',
      from_='+12202723418',
      to='+916382110646'
     )

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

SM04531c8c26b16916c76378541c2b85db
Fire Detected
SMS Sent
```