PROJECT REPORT

Emerging Methods for Early Detection of Forest Fires

submitted by

TEAM ID - 31445-1660200428

TEAM LEADER - K.PRAKASHRAJ
TEAM MEMBER 1 – C.NARESH
TEAM MEMBER 2 – K.THEENA
TEAM MEMBER 3 – B.ILAIYARAJA

TABLE OF CONTENTS

1 INTRODUCTION

PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

Forest fire detection is the ability of computer systems to recognise Fire from various region of forest , such as fire, smoke, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in Detecting the forest fire.

PURPOSE

The main aim of our project is detection and monitoring the forest fire To minimize the effect of fire breakout by controlling in its early stage also to protect Domestic by informing about the breakout to the respective forest department as early as possible . We have implemented the IOT technology to achieve our objective.

2 LITERATURE SURVEY

EXISTING PROBLEM

Some of the relevant literary works in this field are briefed below:

The one fourth area of Karnataka is covered by forest, the forest and bio-diversity of the India are at the considerable chance and beneath enormous pressure. General causes of forest fire are extreme hot and aired weather, lightning and human carelessness. In order to protect these huge stretches of forest land, there need to be taken early caution measures to control of spreading fire

REFERENCES

1. **A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing**

- P. Barmpoutis, P. Papaioannou, K. Dimitropoulos, N. Grammalidis
- Environmental Science
- Sensors
- 2020

An overview of the optical remote sensing technologies used in early fire warning systems is presented and an extensive survey on both flame and smoke detection algorithms employed by each technology is provided.

2. Forest Fire Detection System using LoRa Technology

- N. Gaitan, Paula Hojbota
- Environmental Science
- 2020

This paper proposes a system capable of quickly detecting forest fires on long wide distance using LoRa (Long Range) technology based on LoRaWAN (Long Range Wide Area Network) protocol which is capable to connect low power devices distributed on large geographical areas.

3. **Low Cost LoRa based Network for Forest Fire Detection**

- Roberto Vega-Rodríguez, Sandra Sendra, Jaime Lloret, Pablo Romero-Díaz, José Luis GarcíaNavas
- Computer Science, Environmental Science

  2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)
- 2019

A low cost Long Range (LoRa) based network able to evaluate level of fire risk and the presence of a forest fire and the evaluation algorithm is based on the 3030-30 rule.

4. **A Survey of Machine Learning Algorithms Based Forest Fires Prediction and Detection Systems**

- F. Abid
- Environmental Science, Computer Science
- Fire Technology
- 2020

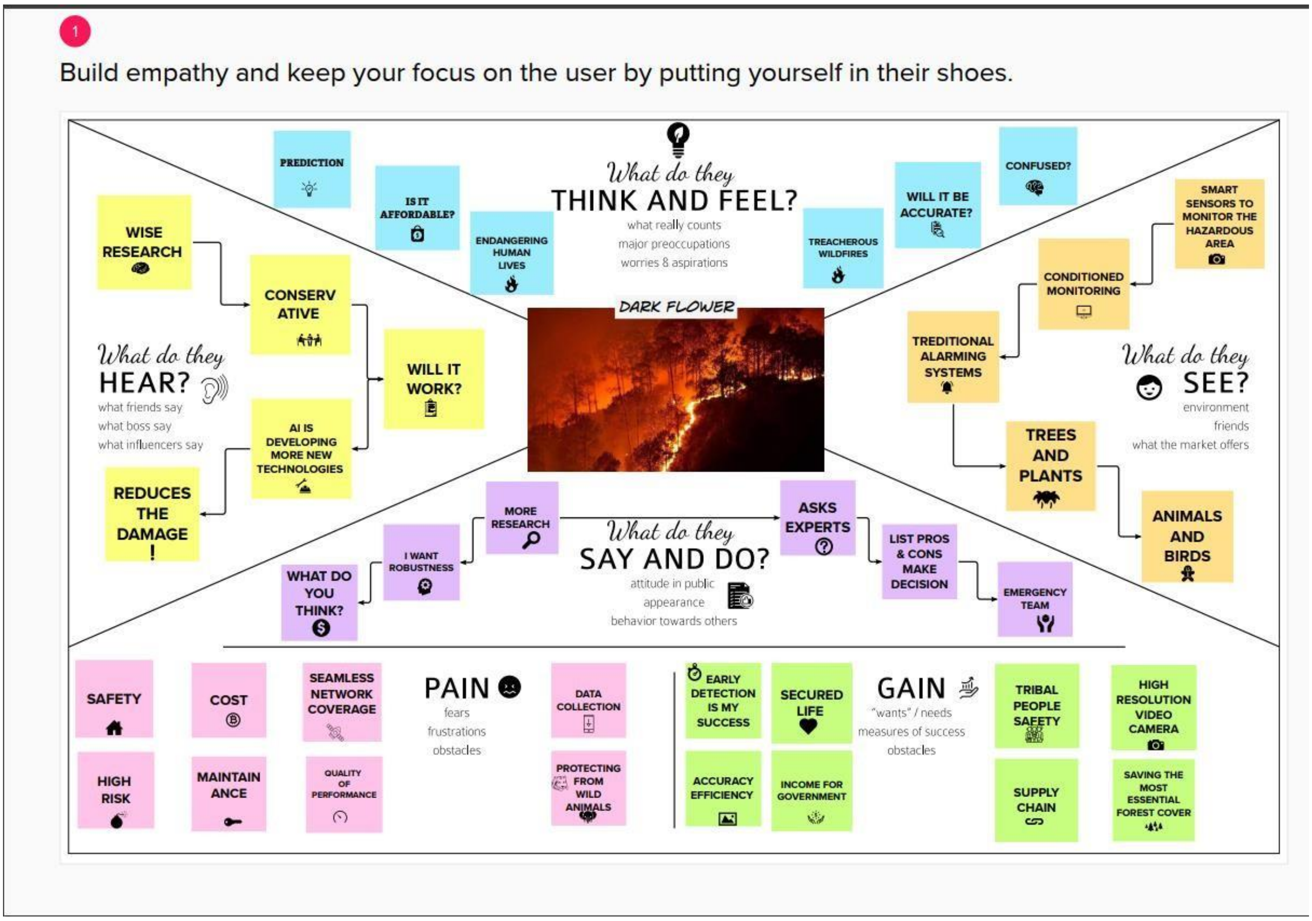
A comprehensive survey of the machine learning algorithms based forest fires prediction and detection systems is presented, highlighting the main issues and outcomes within each study.

2.3 PROBLEM STATEMENT DEFINITION

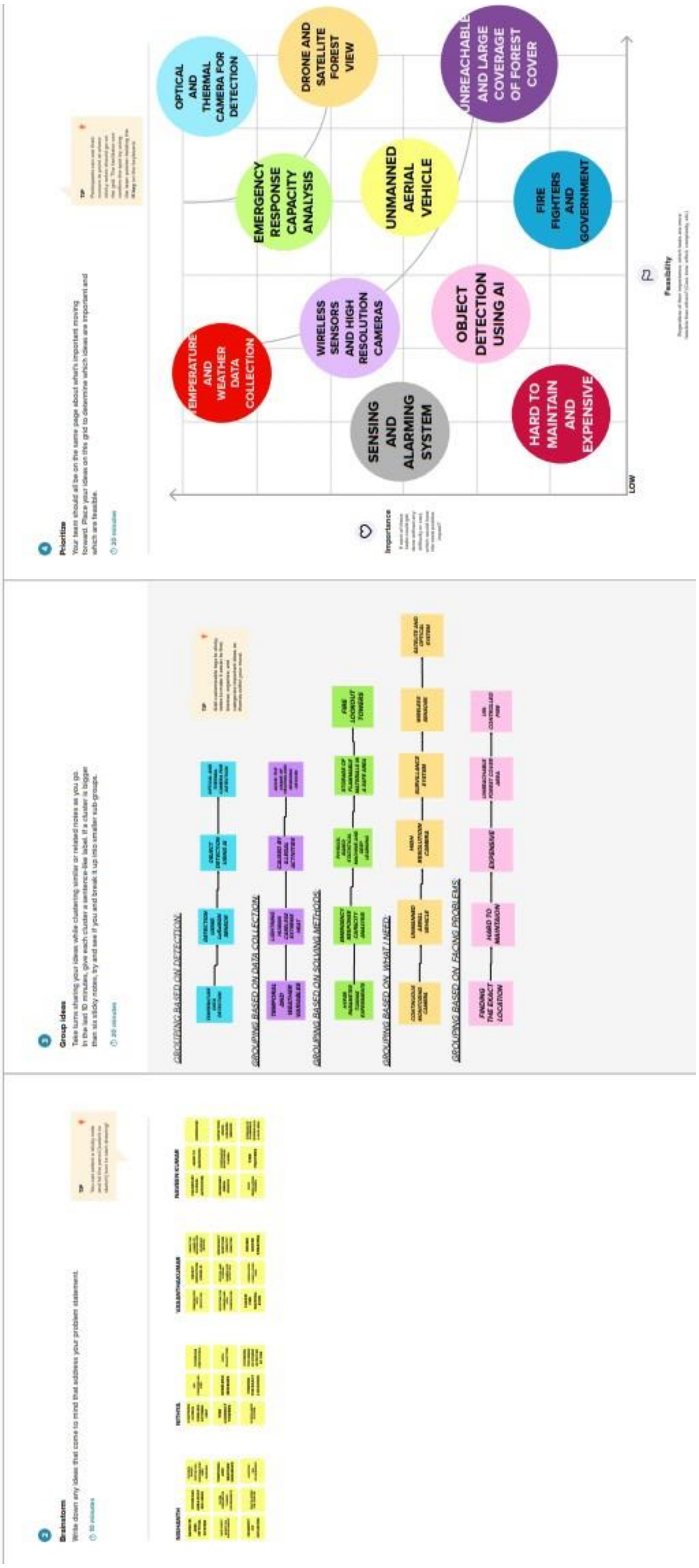
Some people know about the current issues are the most important ones because it is mostly a lot in the news but sometimes other big issues that change our lives are not mentioned in the news because they are issues that can hurt us in the long run or not really important for the modern public. One issue I can tell you about is the forest fires. Sometimes people don 't notice or now about the forest fires until it is talk in the news and it 's mostly because it has done a great damage

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS



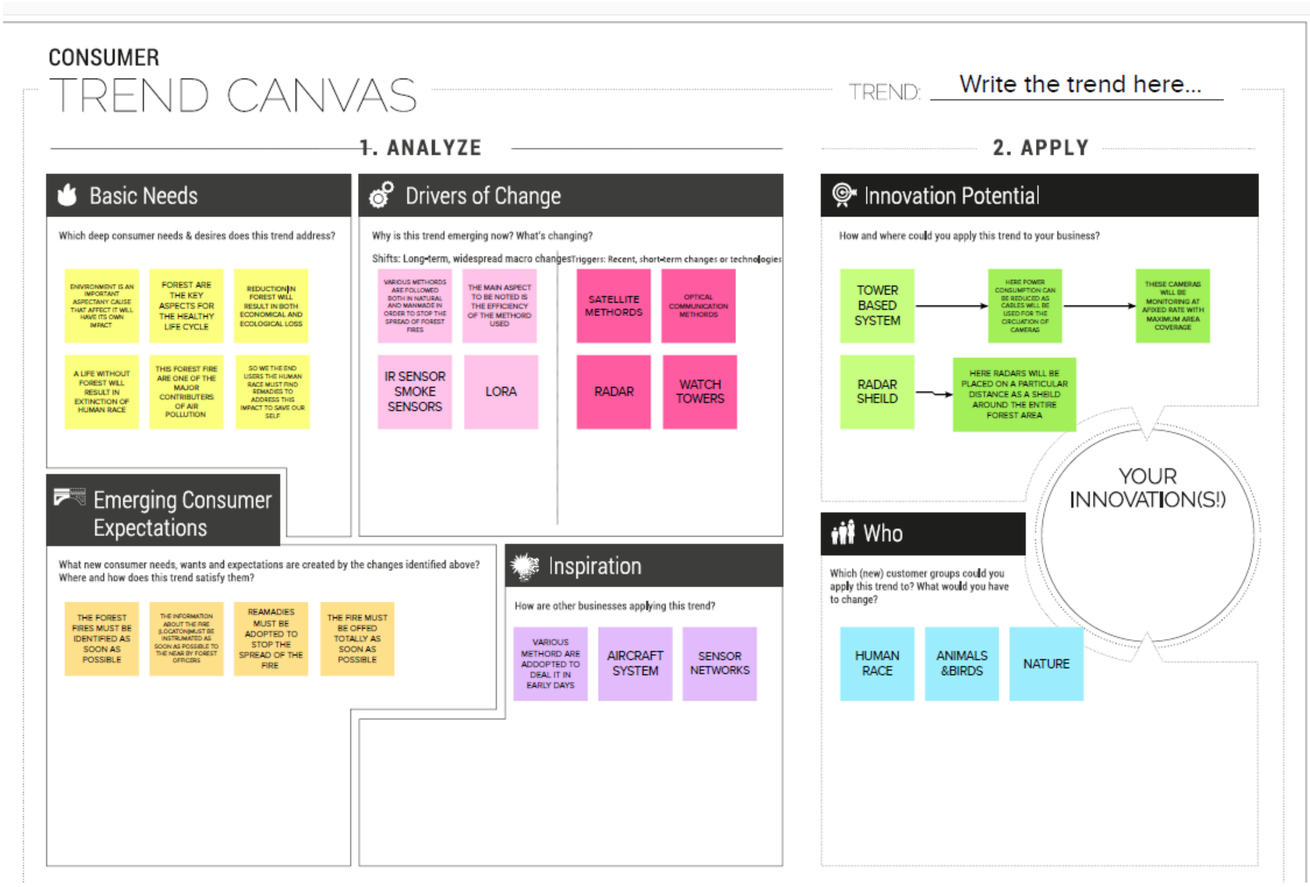
IDEATION & BRAINSTORMING



PROPOSED SOLUTION

Proposed Solution : S.No.	Parameter	Description
1.	Problem Statement	Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities
2.	Idea / Solution description	1 TOWER BASED SYSTEM camera can be moved in cables in order to reduce power consumption and the system can be operated from tall tress which can be interconnected with cables as same as cricket field coverage
3.	Novelty / Uniqueness	Here power consumption can be avoided and constant monitoring over a large area can be obtained
4.	Social Impact / Customer Satisfaction	1 TRIBAL PEOPLE This will result in loss of lives in tribal people and even loss of an community 2ANIMALS This will lead to extinction of various species of animals and push some into endangered species specification 3 HUMANS This will readily affect the economy and ecology of life
5.	Business Model (Revenue Model)	TOWER BASED CABLE INTER CONNECTED CAMERAS
6.	Scalability of the Solution	WE have addressed the field such us constant monitoring as camera will be in movement and the power consumption is also reduced due to cable based interconnection

PROBLEM SOLUTION :



REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS:

-Following are the functional requirements of the proposed solution

Sn. No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1.	User Registration	Registration through G-mail.
2.	User Confirmation	Confirmatio through OTP. Confirmation through mail.
3.	Reporting	User receives a SMS if forest fires occurs.
4.	Detection of Forest Fires	Detects Forest Fires at the earliest.
5.	Video Recording	Records the Forest Footage 24/7

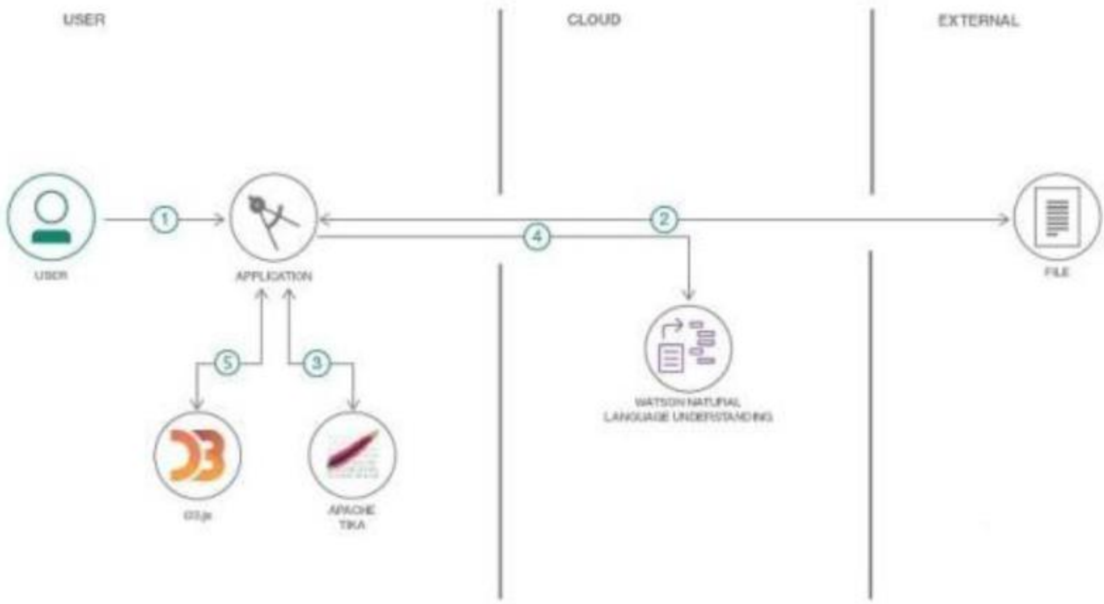
NON-FUNCTIONALREQUIREMENTS:

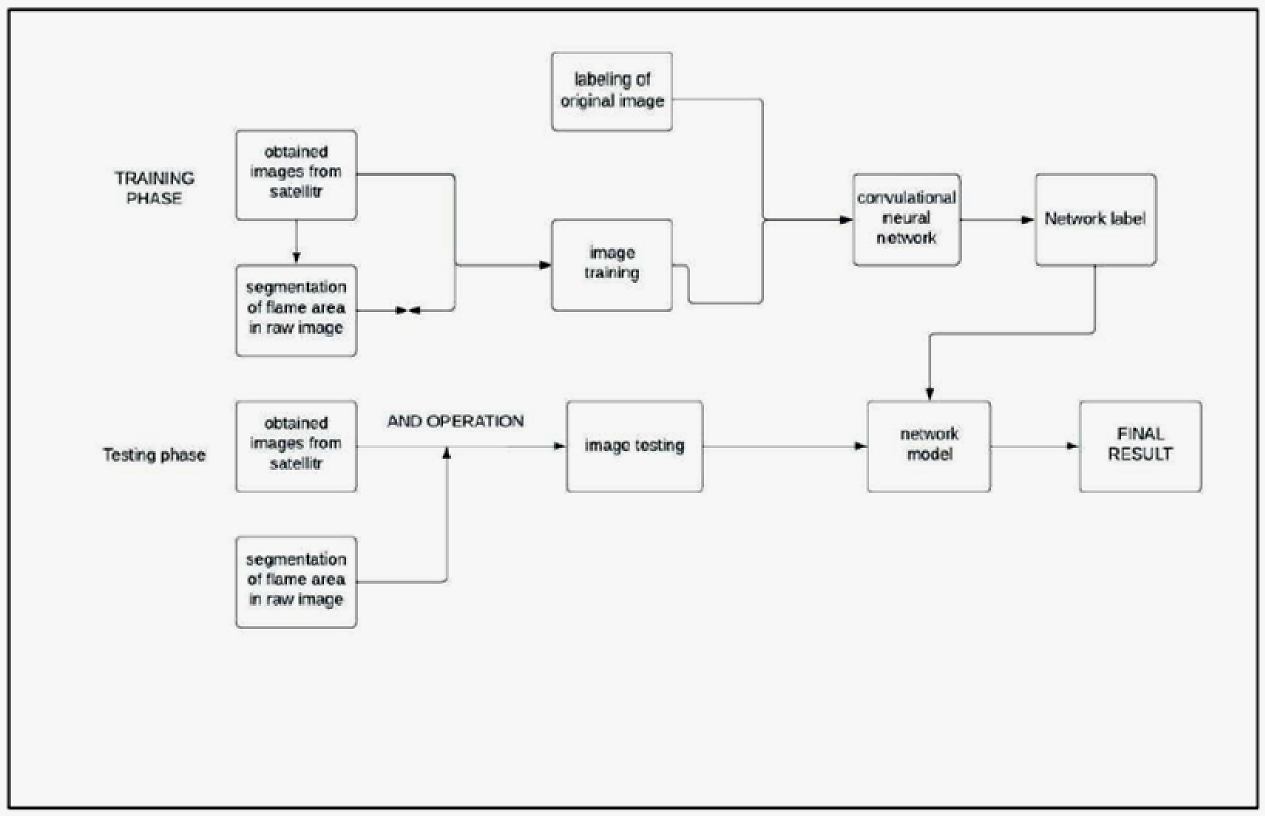
_
__-Following are the non-functional requirements of the proposed solution.

Sn. No.	Non-Functional Requirement	Description
1.	Usability	A non technical person can easily use the App.
2.	Security	Login to the app provides Security.
3.	Reliability	Software updates will be done periodically.
4.	Performance	The respose from the app will be spontaneous.
5.	Availability	The App will be available at all times except during the server maintenance.
6.	Scalability	The Website traffic limit will be 100 users at a time.

PROJECT DESIGN

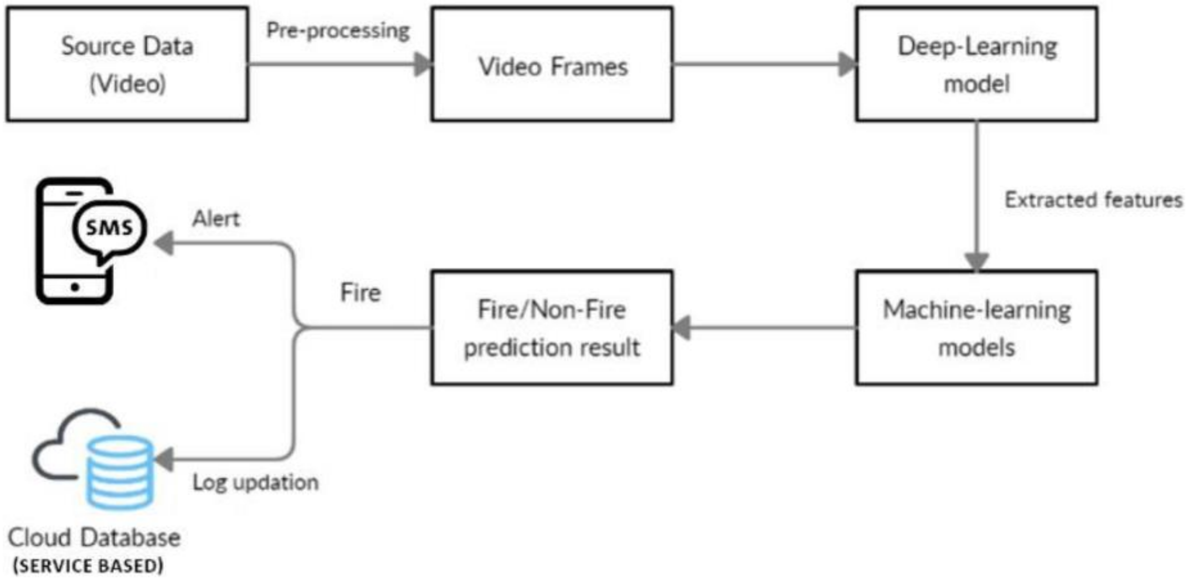
Data Flow Diagram



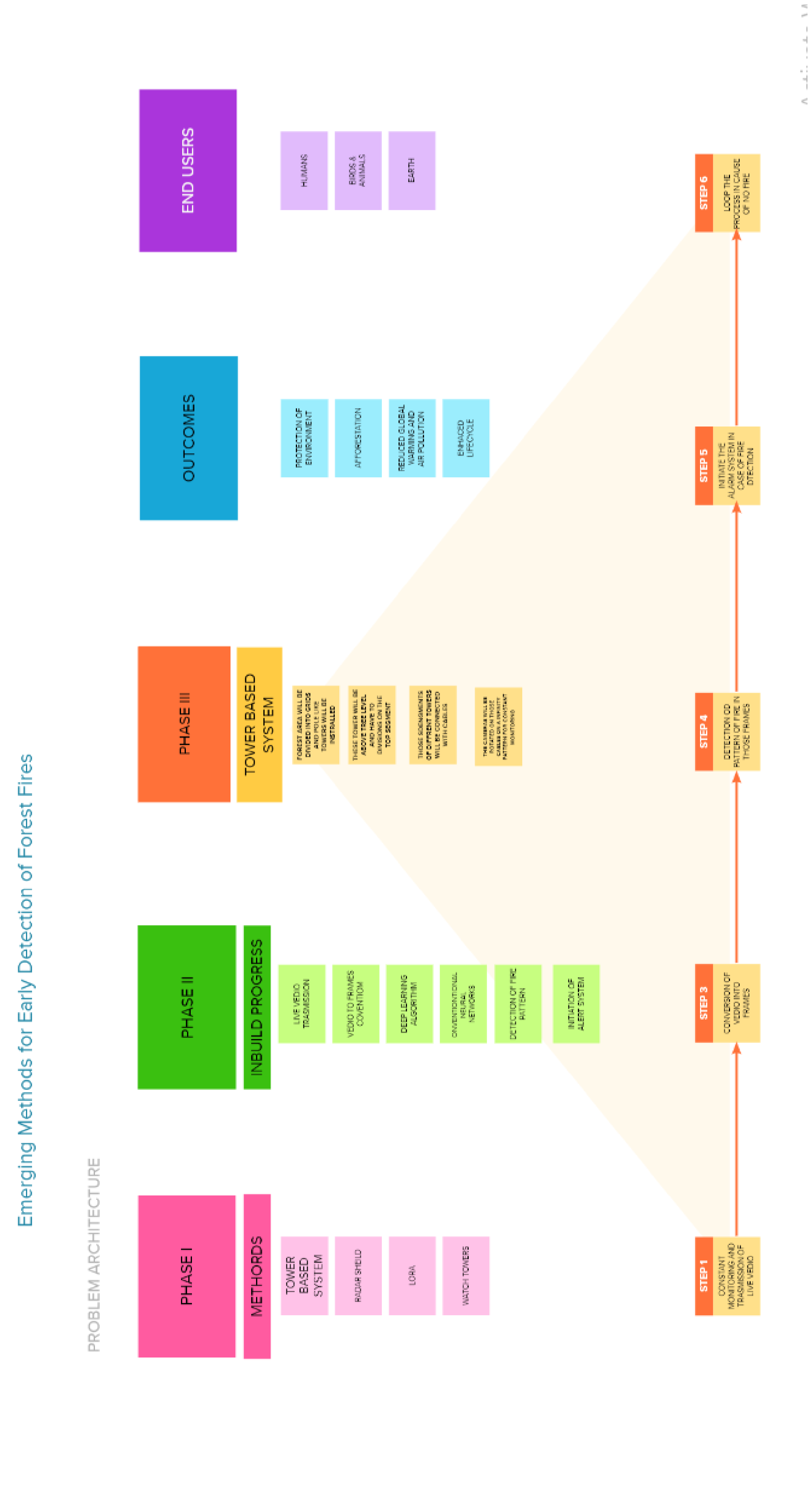


TRAINING PHASE
obtained images from satellitr
segmentation of flame area in raw image
labeling of original image
image training
convulational neural network
Network label
Testing phase
obtained images from satellitr
AND OPERATION
image testing
network model
FINAL RESULT
segmentation of flame area in raw image

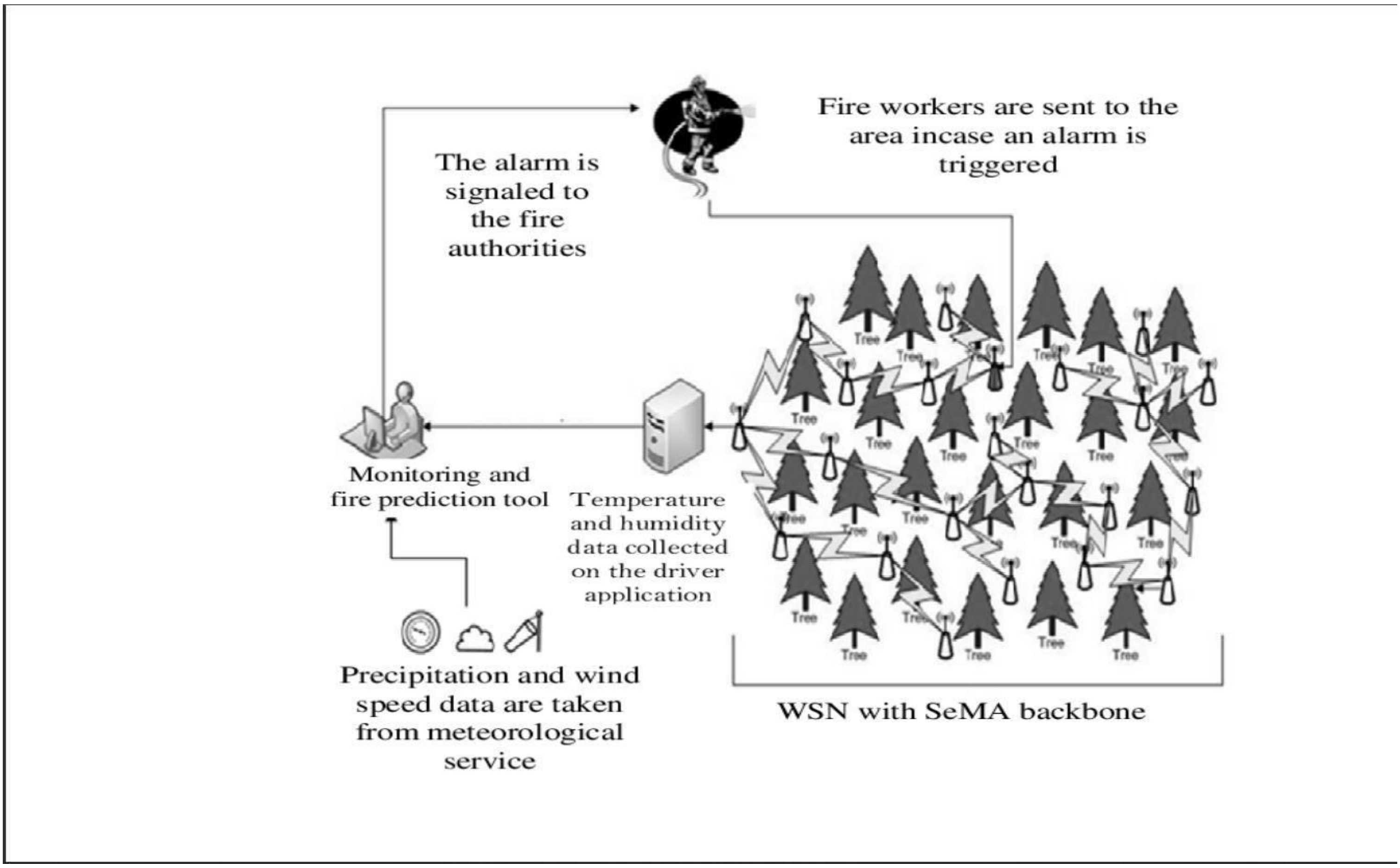
SOLUTION & TECHNICAL ARCHITECTURE :

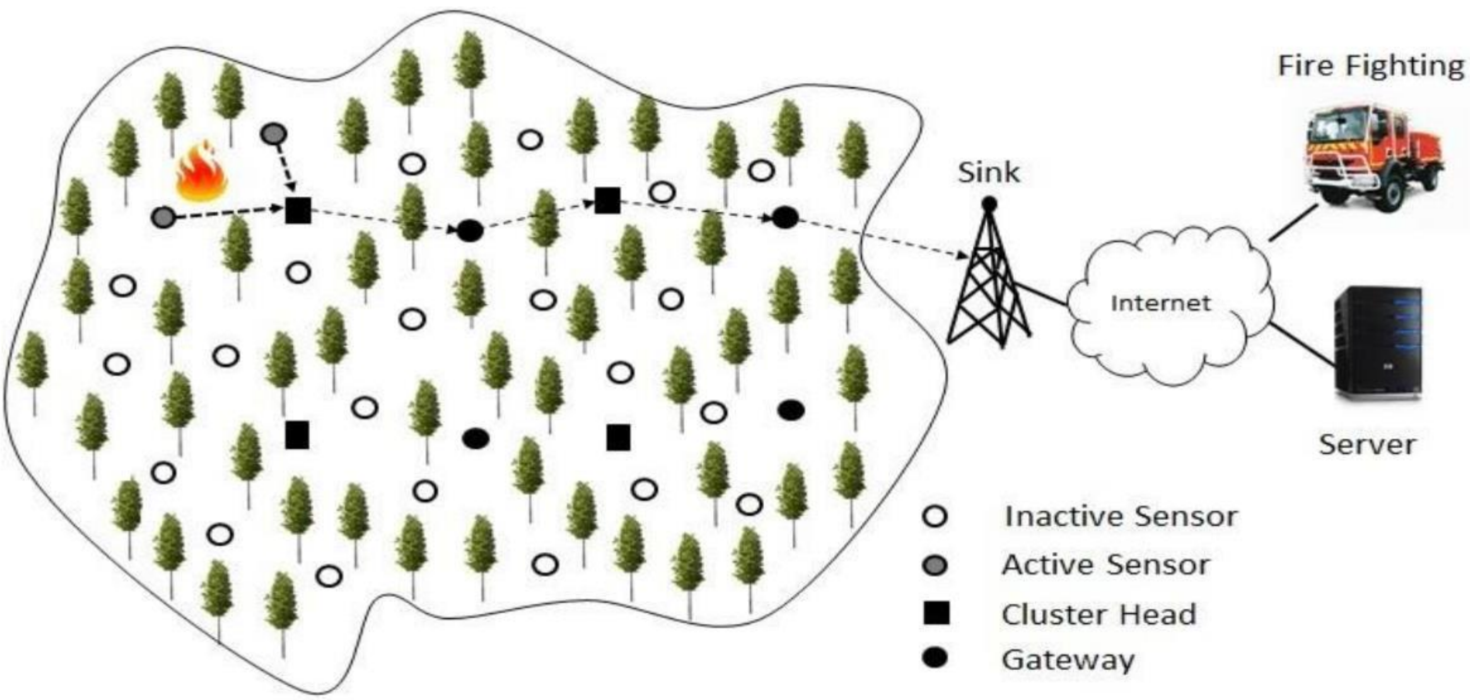


SOLUTION ARCHITECTURE



TECHNICAL ARCHITECTURE





USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Forest Management Team	Setting up a camera	USN-1	As a user, the forest management team has to survey the forest by adding camera to the fire prone areas.	The live video captured can be monitored	High	Sprint-1
		USN-2	As a user, the forest management team can get video feed which is used for processing	The camera sends video or image to the forest centre	High	Sprint-2
		USN-3	Along with forest team, the NGO can also get access of the video to take some early measurement of forest fires.	They can also get the view of the live monitoring of forest	Low	Sprint-1
Technical Team	Image Classification	USN-4	By using CNN Model, the images captured by the camera is classified accordingly by testing & training the model	The model should be able to identify the difference between fire and a normal smoke	Medium	Sprint-2
	Using Open CV	USN-5	The recorded video is under monitoring continuously to determine the detection of early video	Therefore, by using CNN we can determine the input layer, classify the hidden layers and send warnings through output layer	High	Sprint-2
Alert Team	Dashboard	USN-6	Thus, after successful detection of fire by processing images. This, API sends the alert by buzzing the alarm and sends messages through chatbot	Thus, the immediate response which is required for earlier determination through sending quick responses	High	Sprint-2

Fire Management	Twilio API		They play the most important role to cool the	They take the following measures to	High	Sprint-2
			fire and manage the excess spread of fire further	stop fire from spreading		
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Media & Nearby Residing People	News, Radio, Alerts,	USN-7	Protecting wildlife, human from the disaster caused	Thus, helping unit should be sent to protect lives	Medium	Sprint-2

PROJECT PLANNING AND SCHEDULING

SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Pre-processing	USN -1	Data is unclean and contains a lot of invalid data and null values.These are removed.	2	High	Prakashraj.K, Naresh.C, Theena.K.
Sprint-2	Image Augmentation	USN -2	Image in the dataset are divided into different images by changing their size and turned around various degree for improve the learning of the model.	1	High	Ilaiyaraja.B, Prakashraj.K.
Sprint-2	Model creation	USN -3	A articificial neutral network model is Created.	2	Medium	Ilaiyaraja.B, Naresh.C, Theena.K.
Sprint-1	Model Compilation	USN -4	After the pre-processing the model is compiled.	2	Medium	Prakashraj.K, Naresh.C.
Sprint-1	Training model	USN -5	A lot of data is fed to the model and it's trained against various cases.	1	High	Ilaiyaraja.B, Theena.K.
Sprint -2	Testing Model	USN-6	A random images is fed into the model to see if the accuracy is above 75%	2	High	Prakashraj.K, Naresh.C, Theena.K. Ilaiyaraja.B,

SPRINT DELIVERY SCHEDULE

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	‘ 20	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	15	08 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	16 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	5	\ 23 Nov 2022

CODING & SOLUTIONING

```python
import cv2
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from twilio.rest import Client
from playsound import playsound
from decouple import config


message_sent = False

model = load_model("./model.h5")

video = cv2.VideoCapture("fire.mp4")

name = ["No fire", "Fire Detected"]


def send_message():
    account_sid = config("ACCOUNT_SID")
    auth_token = config("AUTH_TOKEN")

    client = Client(account_sid, auth_token)
    message = client.messages.create(
        body="Forest Fire detected , Stay safe!!!",
        from_=config("FROM"),
        to=config("TO")
    )
    print(message.sid)
    print("Fire Detected")
    print("SMS Sent!")

playsound("./beep.mp3")
```

```
while True:
    success, frame = video.read()
    cv2.imwrite("image.jpg", frame)
    img = image.load_img("image.jpg", target_size=(128, 128))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    pred = model.predict(x)
    p = int(pred[0][0])
    cv2.putText(frame, str(name[p]), (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 1)

    if p == 1:
        if not message_sent:
            send_message()
            message_sent = True
        print("Fire Detected , stay safe!!!")
    else:
        print("No Fire Detected")

    cv2.imshow("Image", frame)

    if cv2.waitKey(1) & 0xFF == ord('x'):
        break

video.release()
cv2.destroyAllWindows()
```

USER ACCEPTANCE TESTING

DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	1	1	0	3
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	3	1	0	1	5
Not Reproduced	0	0	0	1	1
Skipped	1	0	1	0	2
Won't Fix	1	0	0	0	1
Total	6	3	4	3	14

TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	2	8
Security	3	0	2	2
Performance	2	0	1	1

Exception Reporting	3	0	0	3

9 RESULTS

9.1 PERFORMANCE METRICS

Locust Test Report

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

Target Host: http://127.0.0.1:5000/

Script: locust.py

Request Statistics

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

Response Time Statistics

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

Charts
Total Requests per Second
RPS
Failures/s
6
5
4
3
2
1
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Response Times (ms)
Median Response Time
95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
Number of Users
Users
120
100
80
60
40
20
0

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES

The proposed system detects the forest fire at a faster rate compared to existing system. It has enhanced data collection feature. The major aspect is that it reduces false alarm and also has accuracy due to various sensors present. It minimizes the human effort as it works automatically. This is very affordable due to which can be easily accessed. The main objective of our project is to receive an alert message through an app to the respective user.

DISADVANTAGES

The electrical interference diminishes the effectiveness of radio receiver. The main drawback is that it has less coverage range areas.

11 CONCLUSION

This type of system is the first of its kind to ensure no further damage is then to forests when there is fire breakout and immediately a message is sent to the user through the App. Immediate response or early warning to a fire breakout is mostly the only ways to avoid losses and environmental, cultural heritage damages to a great extent. Therefore the most important goals in fire surveillance are quick and reliable detection of fire. It is so much easier to suppress fire while it is in its early stages. Information about progress of fire is highly valuable for managing fire during all its stages. Based on this information the firefighting staff can be guided on target to block fire before it reaches cultural heritage sites and to suppress it quickly by utilizing required firefighting equipment and vehicles. With further research and innovation, this project can be implemented in various forest areas so that we can save our forests and maintain great environment.

12 FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

Additional pump can be added so that it automatically sends water when there is a fire breakout. Also industrial sensors can be used for better ranging and accuracy.

❾ This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

APPENDIX

SOURCE CODE

Import the neccessary libraries

```python
import keras
import tensorflow

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

✓ 1m 18.2s Python

```python
import tensorflow

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Python

```python
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)
```

✓ 0.1s Python

```python
x_train = train_datagen.flow_from_directory(r'./Dataset/train_set/',
                                            target_size=(128, 128),
                                            batch_size=32,
                                            class_mode='binary')
```

✓ 0.6s Python

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

```python
x_test = train_datagen.flow_from_directory(r'./Dataset/test_set/',
                                           target_size=(128, 128),
                                           batch_size=32,
                                           class_mode='binary')
```

✓ 0.1s Python

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

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D, Flatten
```

✓ 0.1s Python

```python
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(150,activation="relu"))
model.add(Dense(1, activation="sigmoid"))
```

✓ 1.8s Python

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

✓ 0.1s

```python
model.fit(x_train, steps_per_epoch=14, epochs=10, validation_data=x_test, validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 64s 4s/step - loss: 3.5440 - accuracy: 0.5665 - val_loss: 0.4052 - val_accuracy: 0.8430
Epoch 2/10
14/14 [==============================] - 23s 2s/step - loss: 0.5222 - accuracy: 0.7431 - val_loss: 0.2283 - val_accuracy: 0.9669
Epoch 3/10
14/14 [==============================] - 23s 2s/step - loss: 0.3097 - accuracy: 0.8647 - val_loss: 0.1622 - val_accuracy: 0.9504
Epoch 4/10
14/14 [==============================] - 22s 2s/step - loss: 0.2392 - accuracy: 0.8945 - val_loss: 0.1137 - val_accuracy: 0.9669
Epoch 5/10
14/14 [==============================] - 23s 2s/step - loss: 0.2125 - accuracy: 0.8968 - val_loss: 0.1337 - val_accuracy: 0.9504
Epoch 6/10
14/14 [==============================] - 23s 2s/step - loss: 0.1922 - accuracy: 0.9243 - val_loss: 0.0887 - val_accuracy: 0.9669
Epoch 7/10
14/14 [==============================] - 23s 2s/step - loss: 0.1773 - accuracy: 0.9266 - val_loss: 0.1454 - val_accuracy: 0.9339
Epoch 8/10
14/14 [==============================] - 21s 2s/step - loss: 0.1678 - accuracy: 0.9427 - val_loss: 0.0835 - val_accuracy: 0.9752
Epoch 9/10
14/14 [==============================] - 24s 2s/step - loss: 0.1733 - accuracy: 0.9243 - val_loss: 0.1079 - val_accuracy: 0.9669
Epoch 10/10
14/14 [==============================] - 25s 2s/step - loss: 0.1647 - accuracy: 0.9335 - val_loss: 0.0716 - val_accuracy: 0.9752

<keras.callbacks.History at 0x1920c974be0>
```

Save the model

```python
model.save("model.h5")
```

Prediction

```python
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
```

Python

```python
model = load_model("model.h5")
```

Python

Reviewing the model

```python
img = image.load_img("forest-fire.jpg")
x = image.img_to_array(img)
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
x = np.expand_dims(res, axis=0)
```

Python

```
pred = model.predict(x)
pred = int(pred[0][0])
pred
```

Python

```
1/1 [==============================] - 1s 524ms/step

0
```

Fire.py (Main file)

```
import cv2
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from twilio.rest import Client
from playsound import playsound
from decouple import config


message_sent = False

model = load_model("./model.h5")

video = cv2.VideoCapture("fire.mp4")

name = ["No fire", "Fire Detected"]


def send_message():
    account_sid = config("ACCOUNT_SID")
    auth_token = config("AUTH_TOKEN")

    client = Client(account_sid, auth_token)
    message = client.messages.create(
        body="Forest Fire detected , Stay safe!!!",
        from_=config("FROM"),
        to=config("TO")
    )
    print(message.sid)
    print("Fire Detected")
    print("SMS Sent!")

playsound("./beep.mp3")
```

```python
while True:
    success, frame = video.read()
    cv2.imwrite("image.jpg", frame)
    img = image.load_img("image.jpg", target_size=(128, 128))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    pred = model.predict(x)
    p = int(pred[0][0])
    cv2.putText(frame, str(name[p]), (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 1)

    if p == 1:
        if not message_sent:
            send_message()
            message_sent = True
        print("Fire Detected , stay safe!!!")
    else:
        print("No Fire Detected")

    cv2.imshow("Image", frame)

    if cv2.waitKey(1) & 0xFF == ord('x'):
        break

video.release()
cv2.destroyAllWindows()
```

GITHUB LINK:

https://github.com/IBM-Project-31445-1660200428