SRI VENKATESWARA COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

EMERGING METHOD FOR EARLY DETECTION OF FOREST FIRES

NALAIYA THIRAN PROJECT REPORT 2022

Submitted by

HARISH RAJA	**2127190501041**
ARAVINDH KRISHNA	**2127190501016**
AVINASH	**2127190501023**
JANARTHANAN	**2127190501047**

Team ID: PNT2022TMID53336

NOVEMBER 2022

TABLE OF CONTENTS

1 INTRODUCTION

Nowadays,the techniques for fire detection in large forest areas are based on satellite images and forest guard posts. However,these methods are not suitable for local measurement of the relevant parameters involved in fire risk and, hence, in its early detection. For this purpose, distributed sensing systems as wireless sensor networks (WSN) can offer a suitable measurement resolution .A WSN consists of several sensing nodes which gather information from the surrounding environment and communicate with each other to send the measured data to a base station for further processing. The most important requirements to develop a WSN node are small form factor, to reduce the visual effect in the area where sensors are distributed. This system provides valuable near real-time and also historical data on the forest fires in Europe, the Middle East and North Africa. EFFIS is currently used and supported with data by 25 EU Member States and many other countries. If we compare these values with the average values of the EFFIS reports for the period 2006-2015, the number of forest fires has decreased by 13327 or almost 20%.

1.1 PROJECT OVERVIEW:

Recently, many solutions of different types have been developed forest fire detection and a fire management system have been successfully developed and applied.It is a fact that wildfire detection and wildfire management must be differentiated in order to understand how to set up new fire detection and containment systems. Therefore, we divide forest fire detection and management systems into different classes. Some studies have combined wildfire suppression and detection methods. A research group has investigated forest fire prevention through root cause analysis. In addition, other studies have looked at private management systems for private forest areas. Although some wildfire management systems include wildfire detection, containment and suppression systems, detection is not their primary focus, so fire detection systems should be considered alone to establish more advanced detection systems. Wildfire detection systems can actually be divided into three main categories Likewise, temperature-based systems can be divided into two distinct categories: based on static and dynamic sensors and based on remote sensors. The latter uses tools such as radar, lidar (light detection and ranging), and sodar (sound detection and ranging). In the literature, many authors focus on early fire detection using image processing techniques on satellite images obtained from forests.

Although all are useful scientific studies and a number have been successfully applied, satellite based systems studies have limitations, including testing limited to private forest areas, high costs and the requirement of a suitable satellite.

1.2 PURPOSE:

Forest fires as of late have been annihilating both for normal biological system, biodiversity and woodland economy. With expanding populace weight and change in worldwide atmosphere situation, there is an expansion in level of fires that are a significant reason for declining Indian woodlands. As indicated by woodland study report of India, 50 % of backwoods regions in nation are fire inclined (going from 50 to 90 % in certain conditions of nation). Around 6 % of the woods are inclined to extreme fire harms. The reason for this planned framework is to manufacture a dependable fire location framework so as to know dynamic status of backwoods temperature in specific conditions. It is about the sensors and dynamic checking framework to dodge a significant fire and genuine harm to woods.

2 LITERATURE SURVEY

TITLE AND AUTHOR	PROPOSED APPROACH
G. Hristov, J. Raychev, D. Kinaneva and P. Zahariev, "Emerging Methods for Early Detection of Forest Fires Using Unmanned Aerial Vehicles and Lorawan Sensor Networks," 2018 28th EAEEIE Annual Conference (EAEEIE), 2018, pp. 1-9, doi: 10.1109/EAEEIE.2018.8534245.	Forest fires are occurring throughout the year with an increasing intensity in the summer and autumn periods. These events are mainly caused by the actions of humans, but different nature and environmental phenomena, like lightning strikes or spontaneous combustion of dried leafs or sawdust, can also be credited for their occurrence. Regardless of the reasons for the ignition of the forest fires, they usually cause devastating damage to both nature and humans.
Kasyap, et al (2022)"Early Detection of Forest Fire Using Mixed Learning Techniques and UAV [1]"Varanasi LVSKB	The proposed work deployed on an onboard UAV uses a mixed deep learning technique composed of YOLOv4 tiny and LiDAR techniques. It has achieved 1.24 seconds of classification time with an accuracy of 91% and an F1 score of 0.91.
Younes Oulad Sayad, et al (2019)"Predictive modeling of wildfires: A new dataset and machine learning Approach"	This paper proposed a methodology to analyze the created dataset in order to predict the occurrence of wildfires in a specific region. Artificial Neural Networks and Support Vector Machines were implemented in "Databricks". The model gave good results for both algorithms (SVM 97.48%, NN 98.32%);

Chi Yuan, et al (2017) "Fire detection using infrared images for UAV-based forest fire Surveillance"	The paper proposed an image processing method for the application to UAV for the automatic detection of forest fires in infrared (IR) images. The algorithm makes use of brightness and motion clues along with image processing techniques based on histogram-based segmentation and optical flow approach for fire pixels detection.
B.C. Arrue, et al (2000)"An intelligent system for false alarm reduction in infrared forest-fire Detection"	The FAR system consists of applying new infrared-image processing techniques and artificial neural networks (ANNs), using additional information from meteorological sensors and from a geographical information database, taking advantage of the information redundancy from visual and infrared cameras through a matching process, and designing a fuzzy expert rule base to develop a decision function.
Vinay Dubey, et al (2018) "Forest Fire Detection System Using IoT and Artificial Neural Network	This paper makes use of Internet of things technology.The early fire detection model has been proposed with the help of the Raspberry Pi microcontroller and required sensors. Centralized server is used for storing the data and analyzing that data. Feed-forward fully connected neural network is used for prediction purposes. Then, an alert message is sent to the admin and to the people within the proximity

2.1 EXISTING PROBLEM:

The existing system for detecting fire are smoke alarms and heat alarms. The main disadvantage of the smoke sensor alarm and heat sensor alarms are that just one module is not enough to monitor all the potential fire prone places. The only way to prevent fire is too cautious at the time. Even if they are installed in every nook and corner, it just is not sufficient for an efficient output consistently. As the number of smoke sensor requirement increase the cost will also increase to its multiple. The proposed system can produce consistent and highly accurate alerts within seconds of accident of the fire. It reduces cost because only one software is enough to power the entire network of surveillance. Research is active on this field by data scientists and machine learning researchers. The real challenge is to minimize the error in detection of fire and sending alerts at the right time. The idea of this research is to fabricate a system through IoT sensors, which is arbitrarily spread in the forest and to make a self-sorted out powerful system between the sensors to cover all the enormous territories in the forest that will used to maintain a strategic distance from the fire harm whenever. The capacity of the sensor is to identify fire in the inclusion region between the time intermission of each 5-10 minutes. At the point when the fire is recognized the entirety of the sensor in the region will be dynamic and order to stop the normal assignment. The concept is to build early fire detector using Arduino which is connected with different IoT sensors. Putting all efforts to develop a smarter system by connecting it to a webpage and monitoring the developed system statistics controlled by the Arduino programming. The use of latest technology can help to prevent the catastrophic accidents in forests. The aim is to early detect the fireplace in forest by considering the several factor like smoke, temperature, humidity, flame and based on the data we get from this programming, the forest department will be able to take an appropriate decision and the rescue team will be able to arrive on time at exact location. Consider, if it is a large region and it produces more carbon monoxide than the ordinary vehicle traffic. Surveillance of the danger areas and an early detection of fireplace can appreciably shorten the response time and additionally decrease the practicable injury as nicely as the fee of firefighting. Known rule applies here: 1 minute – 1 cup of water, 2 minutes - 100 liters of water, 10 minutes - 1000 liters of water. The goal is to notice the fireplace as quicker as possible, its actual localization and early notification to the fire devices. When fire starts then the flammable texture may likewise issues fuel to the hearth focal spot. The spot at that point will expand and more extensive.

2. REFERENCES:

1. Mueller, R. C. et al. Differential tree mortality in response to severe drought: evidence for long-term vegetation shifts. J. Ecol. 93 (2021).

2. Gaylord, M. L. et al. Drought predisposes piñon-juniper woodlands to insect attacks and mortality. New Phytol. 198, 567–578 (2021).

3. McDowell, N. et al. Mechanisms of plant survival and mortality during drought: why do some plants survive while others succumb to drought? New Phytol. 178, 719–739 (2021).

4. Vicente-Serrano, S. M. et al. Response of vegetation to drought time-scales across global land biomes. Proc. Natl Acad. Sci. USA 110, 52–57 (2022).

5. Novick, K., Katul, G., McCarthy, H. & Oren, R. Increased resin flow in mature pine trees growing under elevated CO2 and moderate soil fertility. Tree Physiol. 32, 752–763 (2022).

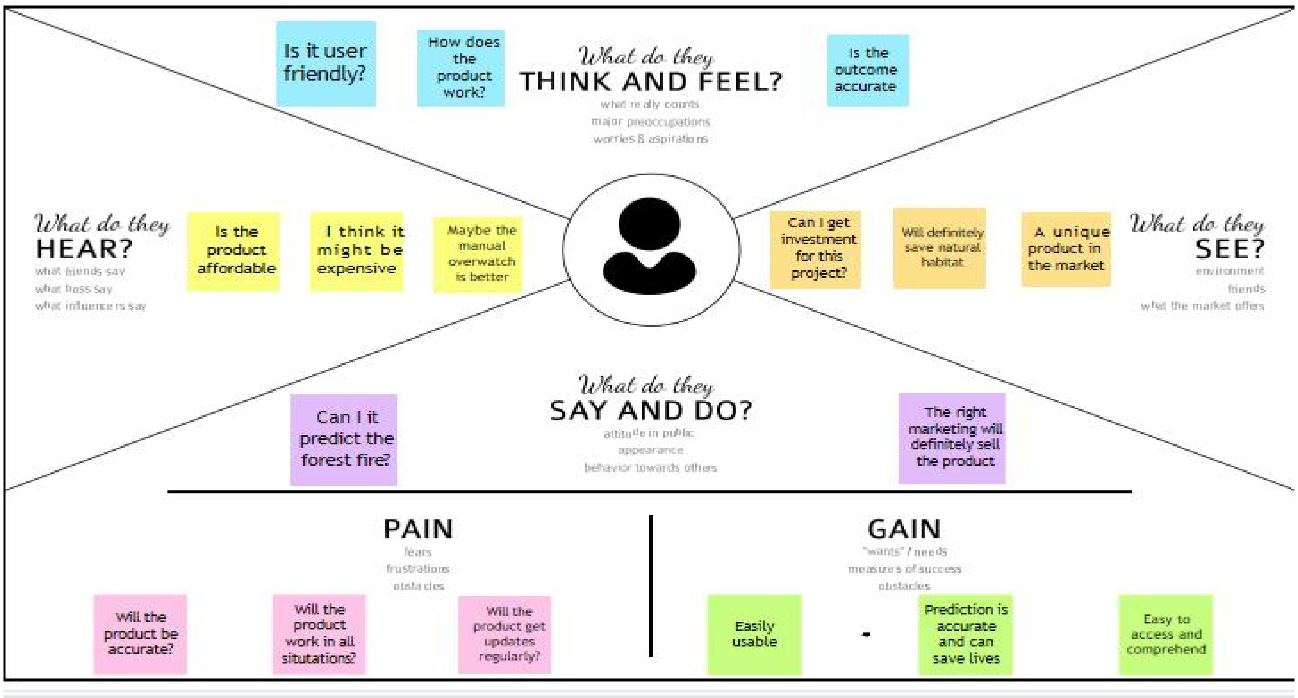
6. Camarero, J. J., Gazol A., Sangesa-Barreda, G., Oliva, J. & Vicente-Serrano, S. M. To die or not to die: early warnings of tree dieback in response to a severe drought. J. Ecol. 103, 44–57 (2022).

2.3 PROBLEM STATEMENT:

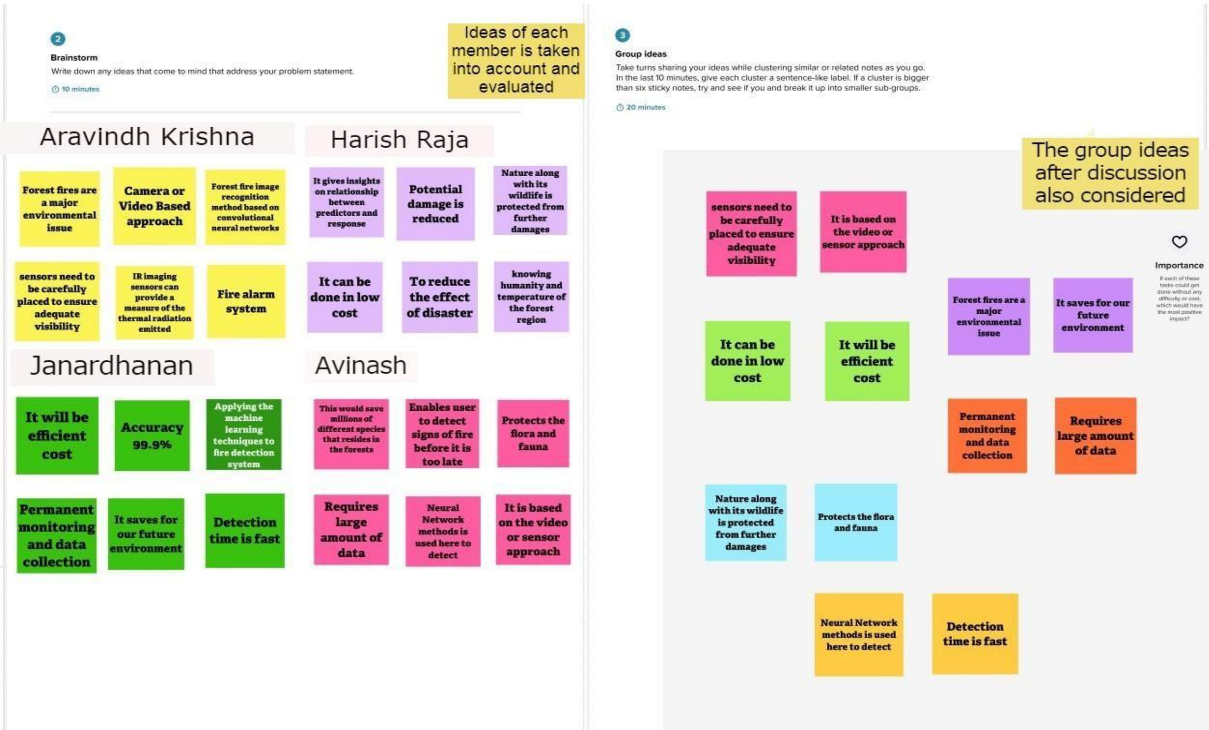
Forest fires are considered as one of the most widespread hazards in a forested landscape. They have a serious threat to forest and its flora and fauna. There are several tangible as well as intangible losses due to forest fires, and they cause an environmental threat to the affected area. Forest fires prediction combines weather factors, terrain, dryness of flammable items, types of flammable items, and ignition sources to analyze and predict the combustion risks of flammable items in the forest. The goal of early detection of forest fires is to predict the fire before it actually occurs. Using Image processing in real time and convolutional neural network, forest fires can be prevented before it happens.

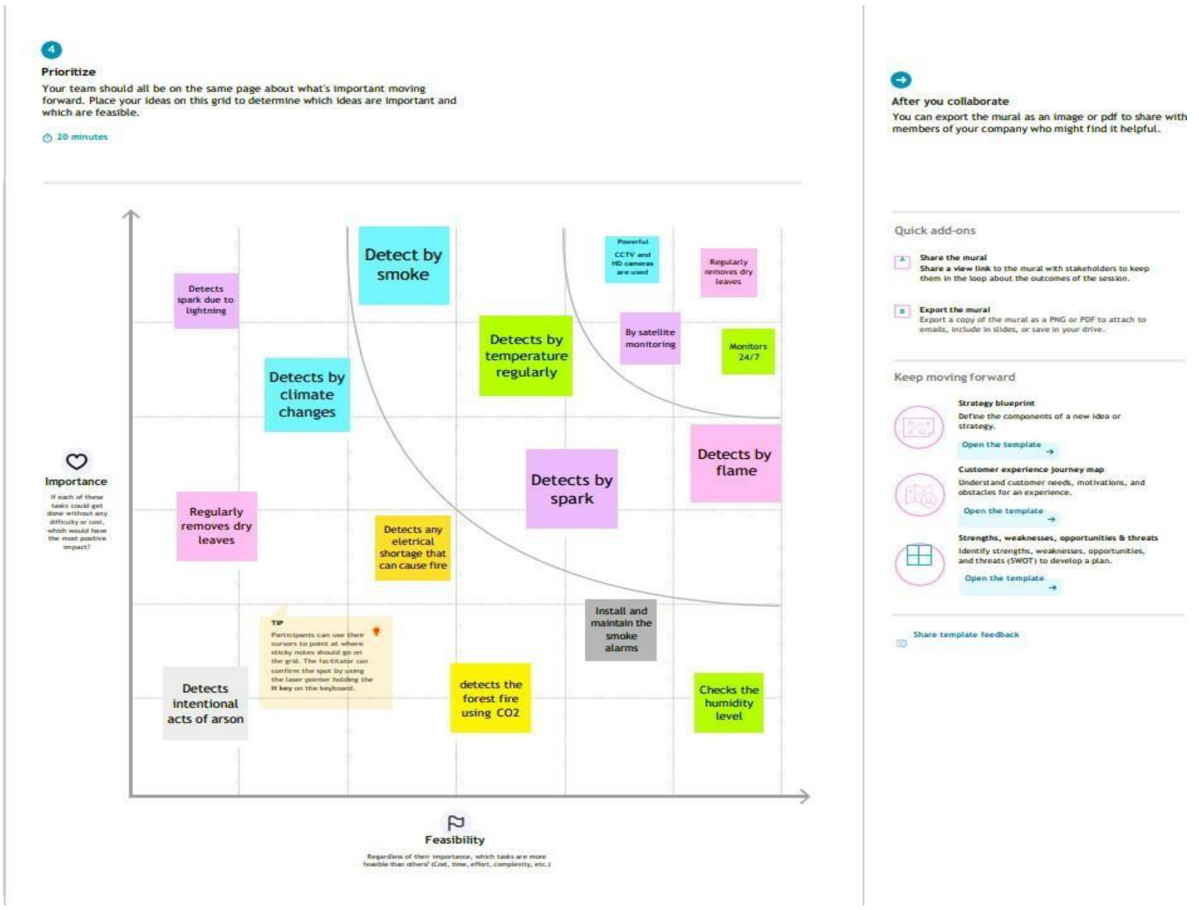
3 IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP:



3.2 IDEATION & BRAINSTORMING:





3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Emerging Methods for Early Detection of Forest Fires
2.	Idea / Solution description	Images collected from satellites and Convolutional neural network.From that images OpenCv methods are used to monitor each frame in the video as an image and predict the forest fire
3.	Novelty / Uniqueness	It is not a ground based method because cameras are not in ground here satellites are used in this method
4.	Social Impact / Customer Satisfaction	This solution will be accepted all over the world because here satellites are used to capture images because of that no ground based cameras are not needed
5.	Business Model (Revenue Model)	It is less cost when compared to the ground based methods like placing sensors all over the forest therefore obtaining satellite images costs less when compared to ground based method
6.	Scalability of the Solution	Scalability of this solution is that it can cover all the forests in the world because satellites can capture forest images all over the world

3.4 PROBLEM SOLUTION FIT:

Define CS, fit into CC	CS	CC	AS	Explore AS, differentiate
	1.Customer Segment(S) • Forest Officials • Wildlife Activists • People who live close to forst	6. CUSTOMER CONSTRAINTS It is difficult to cover an entire forest and to predict fire in a traditional way of overwatch by forest rangers. The budget for manual labor is way too high	5. AVAILABLE SOLUTIONS Outpost across the forest or forest cameras spread across the forest is used to detect forest fires	

Focus on J&P, tap into BE, understand RC

J&P

2.JOBS-TO-BE-DONE / PROBLEMS

Satellite Imaging can help covering over a wide area of trees in a forest to detect fires by the enormous amount of light and heat it produces and a message can be sent to fire fighters via an alert or a message

RC

9.PROBLEM ROOT CAUSE

- Improper discarding of cigarettes
- Lightning
- High wind contributes in spreading of small wires

BE

7. BEHAVIOUR

Customers can't find a perfect solution. So, they prefer Artificial Intelligence.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS

TR

Loss of natural vegetation and destruction of fauna and flora

4. EMOTIONS: BEFORE / AFTER

EM

Before : Hoping and praying that forest fire never comes

After: Assurance on the safety of flora and fauna

10. YOUR SOLUTION

SL

OpenCV method can be used to monitor videos which can be collected from Satellite and Convolutional Neural Network can be used to monitor each frame in the video as an image and predict if the forest fires will happen or not.

8.CHANNELS of BEHAVIOUR

CH

ONLINE : Sensors to detect forest fires placed on random trees throughout the forests

OFFLINE : Awareness Camp and events should be conducted

Identify strong TR & EM

4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

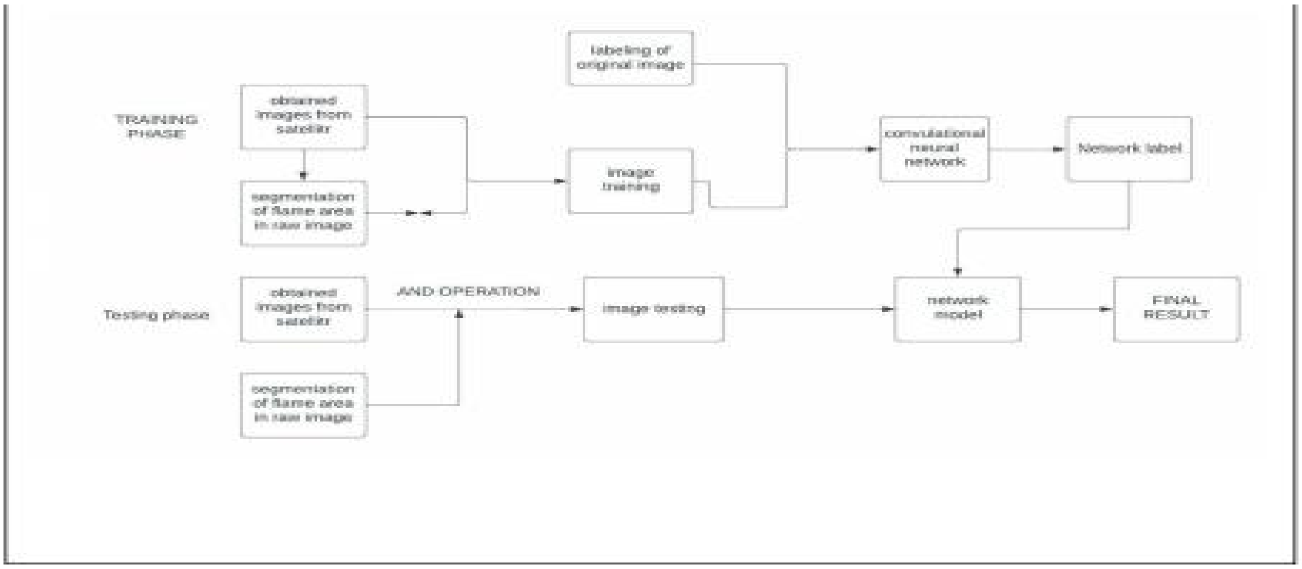
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Reporting	User receives a SMS if forest fires occurs
FR-4	Detection of forest fire	Detects forest fire at the earliest
FR-5	Video Recording	Records the forest footage 24/7

4.2 NON FUNCTIONAL REQUIREMENTS:

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

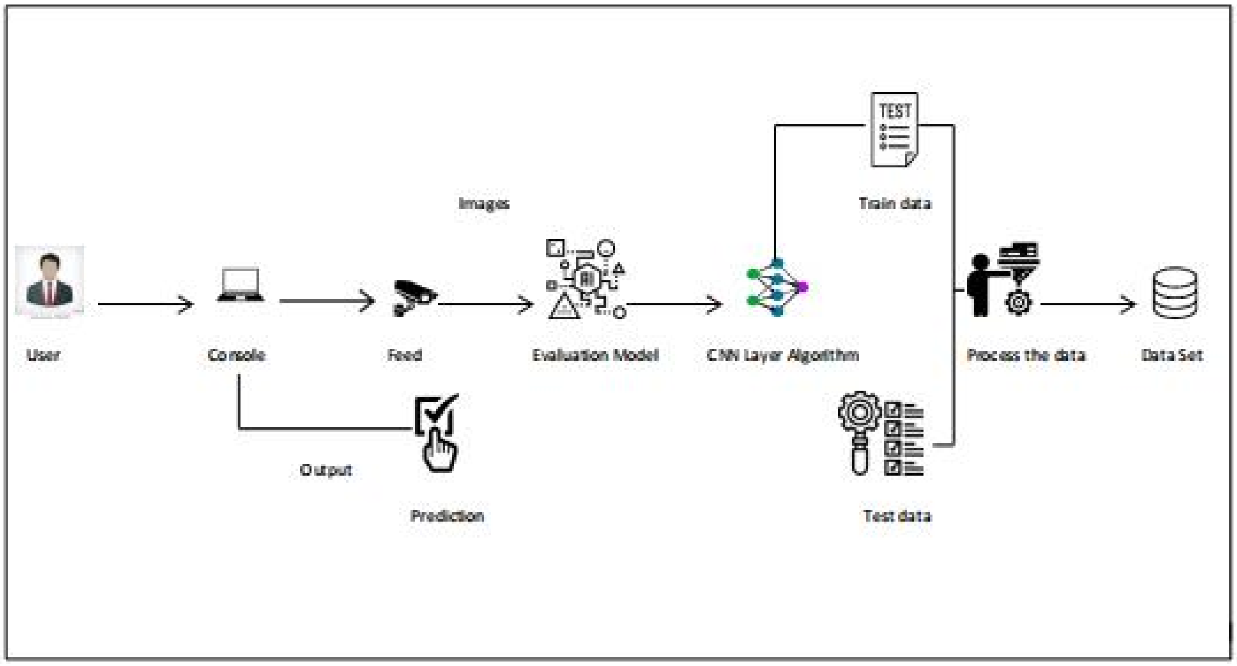
5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:



5.2 SOLUTION &TECHNICAL ARCHITECTURE:

Forest and urban fires have been and still are serious problem for many countries in the world. Currently, there are many different solutions to fight forest fires.These solutions mainly aim to mitigate the damage caused by the fires, using methods for their early detection. Here ,we discuss a new approach for fire detection and control, in which modern technologies are used.we propose a platform that uses Unmanned Aerial Vehicles which constantly patrol over potentially threatened by fire areas.



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email & password	User will get confirmation mail in their registered gmail	High	Sprint-1
Customer (Web user)	Web Registeration	USN-1	User have to register by giving their personal information,gmail,password	User will get confirmation mail in their registered gmail	High	Sprint-1
	login	USN-2	User have to login their account by using their username and password		High	Sprint-1
Administrator	Admin Registeration	USN-1	Admin register through direct verification Admin will get separate username and password		High	Sprint-1

6 PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION:

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	3 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	8 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	16 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	23 Nov 2022

Velocity:

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

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.2 SPRINT DELIVERY SCHEDULE:

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	Aravindh Krishna, Avinash
Sprint-1	Image Augmentation	USN-2	Images in the dataset are divided into different images by changing their size,and turned around various degrees for improve the learning of the model	1	High	Janarthanan, Harish Raja
Sprint-2	Model Creation	USN-3	A Artificial Neural Network Model is Created	2	Medium	Aravindh Krishna, Harish Raja
Sprint-1	Model Compilation	USN-4	After the pre-processing the model is compiled	2	Medium	Avinash, Jamarthanan
Sprint-1	Training Model	USN-5	A lot of data is fed to the model and it's trained against various cases	1	High	Harish Raja, Avinash
Sprint-2	Testing Model	USN-6	A random images is fed into the model to see if the accuracy is above 75%	2	High	Aravindh Krishna, Harish Raja

7 CODING AND SOLUTIONING

7.1 FEATURES

```
In [1]: import zipfile
```

```
In [2]: with zipfile.ZipFile('Dataset.zip','r') as zip_ref:
            zip_ref.extractall()
```

```
In [3]: # importing keras library
        import keras

        # importing ImageDataGenerator class from keras
        from keras.preprocessing.image import ImageDataGenerator
```

```
C:\Users\haris\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required
for this version of SciPy (detected version 1.23.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

```
In [4]: # Defining Parameters and Arguments for ImageDataGenerator Class
        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)
```

```
In [5]: # Applying ImageDataGenerator functionality to trainset
        x_train = train_datagen.flow_from_directory('Dataset/train_set',
                                                    target_size= (128,128),
                                                    batch_size=32,
                                                    class_mode='binary')
```

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

```
In [6]: # Applying ImageDataGenerator functionality to testset
        x_test = test_datagen.flow_from_directory('Dataset/test_set',
                                                  target_size= (128,128),
                                                  batch_size=32,
                                                  class_mode='binary')
```

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

```
In [7]: # Importing Model Building Libraries
        from keras.models import Sequential

        from keras.layers import Dense

        from keras.layers import Convolution2D

        from keras.layers import MaxPooling2D

        from keras.layers import Flatten
```

```
In [8]: import warnings

        warnings.filterwarnings('ignore')
```

```
In [9]: # Initializing The Model
        model = Sequential()
```

```
In [10]: #add Convolutional Layer
         model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

In [11]:
```
# add MaxPooling Layer
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [12]:
```
# add Flatten Layer
model.add(Flatten())
```

In [13]:
```
# add hidden layer
model.add(Dense(150,activation='relu'))
```

In [14]:
```
# add Output Layer
model.add(Dense(1,activation='sigmoid'))
```

In [15]:
```
#Configuring The Learning Process

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

In [16]:
```
# Training The Model
model.fit_generator(x_train,
                    steps_per_epoch=14,
                    epochs=10,
                    validation_data=x_test,
                    validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 21s 1s/step - loss: 2.5009 - accuracy: 0.7087 - val_loss: 0.1858 - val_accuracy: 0.942
1
Epoch 2/10
14/14 [==============================] - 17s 1s/step - loss: 0.3901 - accuracy: 0.8601 - val_loss: 0.1068 - val_accuracy: 0.958
7
Epoch 3/10
14/14 [==============================] - 18s 1s/step - loss: 0.2816 - accuracy: 0.8853 - val_loss: 0.1054 - val_accuracy: 0.942
1
Epoch 4/10
14/14 [==============================] - 18s 1s/step - loss: 0.1907 - accuracy: 0.9106 - val_loss: 0.0940 - val_accuracy: 0.975
2
Epoch 5/10
14/14 [==============================] - 17s 1s/step - loss: 0.1686 - accuracy: 0.9312 - val_loss: 0.0996 - val_accuracy: 0.942
1
Epoch 6/10
14/14 [==============================] - 18s 1s/step - loss: 0.1762 - accuracy: 0.9197 - val_loss: 0.0852 - val_accuracy: 0.958
7
Epoch 7/10
14/14 [==============================] - 17s 1s/step - loss: 0.1728 - accuracy: 0.9220 - val_loss: 0.0573 - val_accuracy: 0.975
2
Epoch 8/10
14/14 [==============================] - 17s 1s/step - loss: 0.1838 - accuracy: 0.9358 - val_loss: 0.0464 - val_accuracy: 0.991
7
Epoch 9/10
14/14 [==============================] - 17s 1s/step - loss: 0.2046 - accuracy: 0.8968 - val_loss: 0.0491 - val_accuracy: 0.983
5
Epoch 10/10
14/14 [==============================] - 19s 1s/step - loss: 0.1873 - accuracy: 0.9151 - val_loss: 0.0950 - val_accuracy: 0.966
9
```

Out[16]:
```
<keras.callbacks.History at 0x29f78ef53a0>
```

7.2 FEATURE 2

```
In [17]: # Save The Model
         model.save("forest1.f5")

         WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op while saving (showing 1 of 1). These functions will
         not be directly callable after loading.

         INFO:tensorflow:Assets written to: forest1.f5\assets

         INFO:tensorflow:Assets written to: forest1.f5\assets

In [18]: #importing necessary libraries for predictions
         from keras.models import load_model

         from tensorflow.keras.preprocessing import image

         import numpy as np

         import cv2

In [19]: # load the saved model
         model = load_model("forest1.f5")

In [20]: #random image path
         img=image.load_img('Dataset/test_set/with fire/180802_CarrFire_010_large_700x467.jpg')
         x=image.img_to_array(img)
         res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
         #expand the image shape
         x=np.expand_dims(res,axis=0)
```

```
In [21]: pred = model.predict(x)
         pred

         1/1 [==============================] - 0s 191ms/step

Out[21]: array([[1.]], dtype=float32)

In [22]: pip install twilio

         Requirement already satisfied: twilio in c:\users\haris\anaconda3\lib\site-packages (7.15.2)
         Requirement already satisfied: requests>=2.0.0 in c:\users\haris\anaconda3\lib\site-packages (from twilio) (2.26.0)
         Requirement already satisfied: pytz in c:\users\haris\anaconda3\lib\site-packages (from twilio) (2021.3)
         Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in c:\users\haris\anaconda3\lib\site-packages (from twilio) (2.1.0)
         Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\haris\anaconda3\lib\site-packages (from requests>=2.0.0->t
         wilio) (2.0.4)
         Requirement already satisfied: certifi>=2017.4.17 in c:\users\haris\anaconda3\lib\site-packages (from requests>=2.0.0->twilio)
         (2022.9.24)
         Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\haris\anaconda3\lib\site-packages (from requests>=2.0.0->twili
         o) (1.26.7)
         Requirement already satisfied: idna<4,>=2.5 in c:\users\haris\anaconda3\lib\site-packages (from requests>=2.0.0->twilio) (3.2)
         Note: you may need to restart the kernel to use updated packages.

         WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
         WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
         WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
         WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
         WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
         WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
```

In [23]:
```
pip install playsound
```

```
Requirement already satisfied: playsound in c:\users\haris\anaconda3\lib\site-packages (1.3.0)
Note: you may need to restart the kernel to use updated packages.
```

```
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
```

In [24]:
```
pip install pygobject
```

```
Collecting pygobject
  Using cached PyGObject-3.42.2.tar.gz (719 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
    Preparing wheel metadata: started
    Preparing wheel metadata: finished with status 'done'
Collecting pycairo>=1.16.0
  Using cached pycairo-1.22.0-cp39-cp39-win_amd64.whl (1.3 MB)
Building wheels for collected packages: pygobject
  Building wheel for pygobject (PEP 517): started
  Building wheel for pygobject (PEP 517): finished with status 'error'
Failed to build pygobject
Note: you may need to restart the kernel to use updated packages.
```

```
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cikit-learn (c:\users\haris\anaconda3\lib\site-packages)
  ERROR: Command errored out with exit status 1:
   command: 'C:\Users\haris\anaconda3\python.exe' 'C:\Users\haris\anaconda3\lib\site-packages\pip\_vendor\pep517\in_process\
```

In [25]:
```
# Importing Libraries

import cv2

import numpy as np

from keras.preprocessing import image

from keras.models import load_model

from twilio.rest import Client

from playsound import playsound
```

```
  Arguments: ('build\\lib.win-amd64-cpython-39\\pygtkcompat',)
  --- Logging error ---
  Traceback (most recent call last):
    File "C:\Users\haris\anaconda3\lib\logging\__init__.py", line 1086, in emit
      stream.write(msg + self.terminator)
  ValueError: underlying buffer has been detached
  Call stack:
    File "C:\Users\haris\anaconda3\lib\site-packages\pip\_vendor\pep517\in_process\_in_process.py", line 349, in <module>
      main()
    File "C:\Users\haris\anaconda3\lib\site-packages\pip\_vendor\pep517\in_process\_in_process.py", line 331, in main
      json_out['return_val'] = hook(**hook_input['kwargs'])
    File "C:\Users\haris\anaconda3\lib\site-packages\pip\_vendor\pep517\in_process\_in_process.py", line 248, in build_wheel
      return _build_backend().build_wheel(wheel_directory, config_settings,
    File "C:\Users\haris\AppData\Local\Temp\pip-build-env-jnofcf1x\overlay\Lib\site-packages\setuptools\build_meta.py", line
412, in build_wheel
      return self._build_with_temp_dir(['bdist_wheel'], '.whl',
    File "C:\Users\haris\AppData\Local\Temp\pip-build-env-jnofcf1x\overlay\Lib\site-packages\setuptools\build_meta.py", line
397, in _build_with_temp_dir
```

```
In [26]: # load the saved model
         model = load_model(r'forest1.f5')

In [27]: #define video
         video = cv2.VideoCapture(0)

In [28]: #define the features
         name = ['forest','with fire']

In [30]: # twilio account sid
         account_sid = 'AC5e0a0a1b5c0b23fe59220dd0c8372c80'
         # twilio account authentication token
         auth_token = '47544858b3b218b1dc7ad63d321afc42'

         client = Client(account_sid, auth_token)
         message = client.messages\
         .create(
             body ='Forest Fire is detected, stay alert',
             #using twilio free number
             from_='+1 479 397 4248',
             #to number
             to='+91 73587 24469')
         print(message.sid)

         SM87620bfcd27607d1398d4d25b64d05a3

In [31]: import tensorflow as tf
```

```
In [31]: import tensorflow as tf

In [32]: import cv2

         import numpy as np

         from keras.preprocessing import image

         from keras.models import load_model

         from twilio.rest import Client

         from playsound import playsound

         video = cv2.VideoCapture(0)
         name = ['forest','with fire']

In [33]: video.isOpened()

Out[33]: True

In [34]: from tensorflow.keras.preprocessing import image

In [35]: from IPython.display import Audio
         from IPython import display
```

In [36]:
```
display.Image("Bandipur_fires_2019.jpg")
```

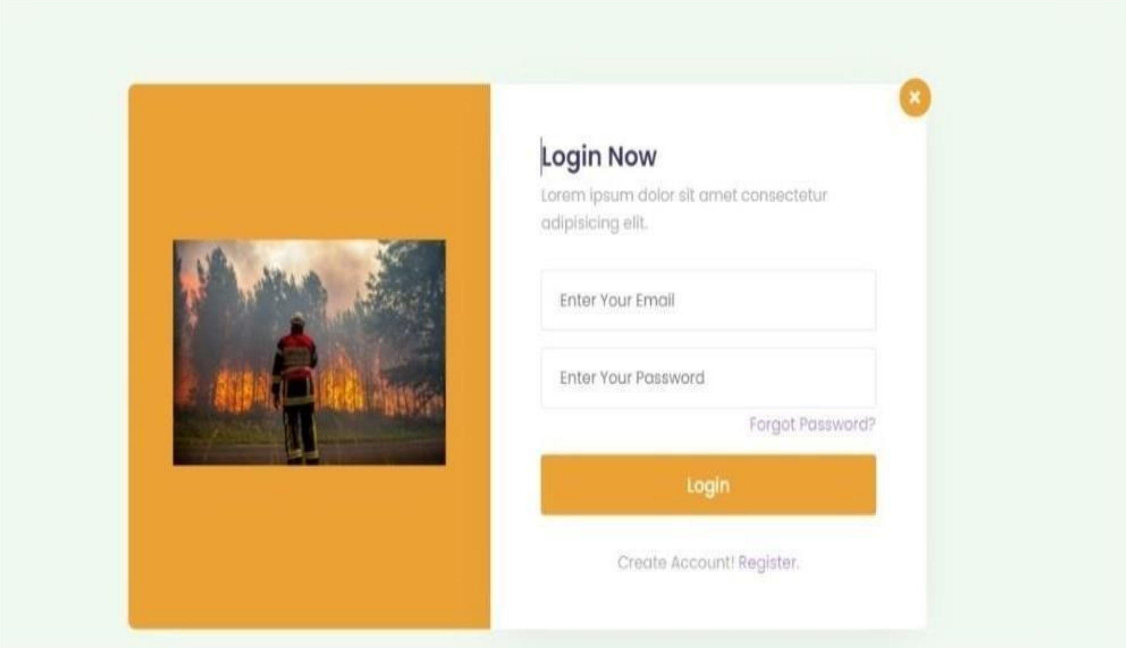
Out[36]:

In [38]:
```
while(video.isOpened()):
  success,frame=video.read()
  cv2.imwrite("Bandipur_fires_2019.jpg.jpg",frame)
  img=image.load_img("Bandipur_fires_2019.jpg",target_size=(128,128))
  x=image.img_to_array(img)
  x=np.expand_dims(x,axis=0)
  pred=model.predict(x)
  p=pred[0]
  print(pred)
  cv2.putText(frame,"predicted class = ",(100,100),cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
  if pred[0]==1:
    account_sid='AC5e0a0a1b5c0b23fe59220dd0c8372c80'
    auth_token='47544858b3b218b1dc7ad63d321afc42'
    client=Client(account_sid,auth_token)
    message=client.messages \
    .create(
        body="Forest fire is detected ,stay alert",
        from_='+1 479 397 4248',
        to='+91 73587 24469')
    print(message.sid)
    print('Fire detected')
    print('SMS sent')
    break
  else:
    print('No danger')
    break
  if cv2.waitKey(1) & 0xFF==ord('a'):
    break
video.release()
cv2.destroyAllWindows()
```

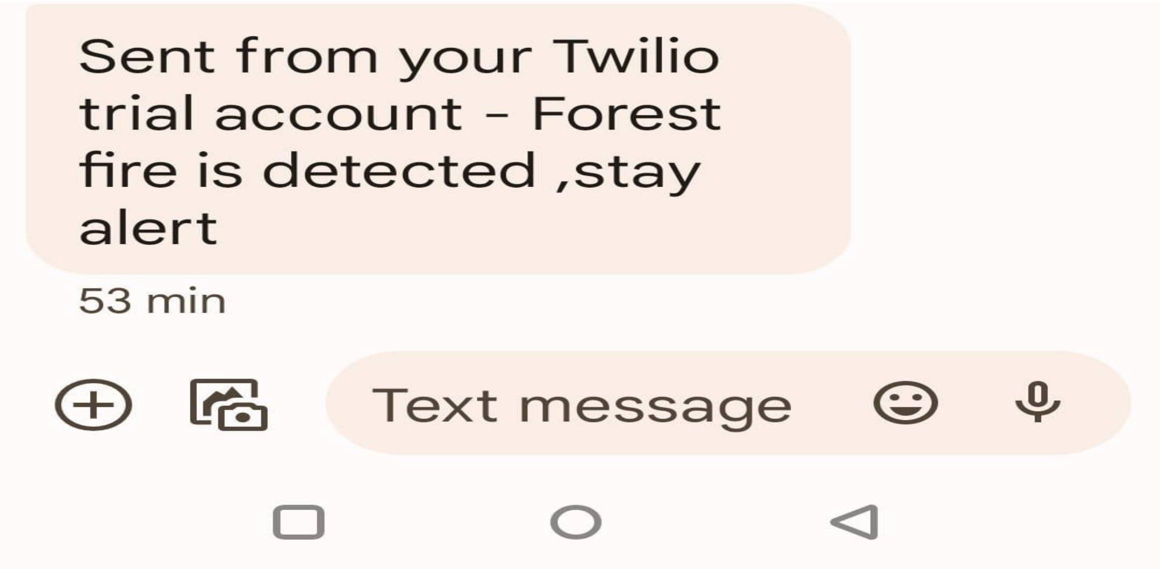
```
1/1 [==============================] - 0s 31ms/step
[[1.]]
SM3e0c10092b627955c0d1884eeb56ad5c
Fire detected
SMS sent
```

8 TESTING

8.1 TEST CASES:



8.2 USER ACCEPTANCE TESTING:



9 RESULTS

9.1 PERFORMANCE METRICS:

```
In [16]: # Training The Model
         model.fit_generator(x_train,
                             steps_per_epoch=14,
                             epochs=10,
                             validation_data=x_test,
                             validation_steps=4)

         Epoch 1/10
         14/14 [==============================] - 21s 1s/step - loss: 2.5009 - accuracy: 0.7087 - val_loss: 0.1858 - val_accuracy: 0.942
         1
         Epoch 2/10
         14/14 [==============================] - 17s 1s/step - loss: 0.3901 - accuracy: 0.8601 - val_loss: 0.1068 - val_accuracy: 0.958
         7
         Epoch 3/10
         14/14 [==============================] - 18s 1s/step - loss: 0.2816 - accuracy: 0.8853 - val_loss: 0.1054 - val_accuracy: 0.942
         1
         Epoch 4/10
         14/14 [==============================] - 18s 1s/step - loss: 0.1907 - accuracy: 0.9106 - val_loss: 0.0940 - val_accuracy: 0.975
         2
         Epoch 5/10
         14/14 [==============================] - 17s 1s/step - loss: 0.1686 - accuracy: 0.9312 - val_loss: 0.0996 - val_accuracy: 0.942
         1
         Epoch 6/10
         14/14 [==============================] - 18s 1s/step - loss: 0.1762 - accuracy: 0.9197 - val_loss: 0.0852 - val_accuracy: 0.958
         7
         Epoch 7/10
         14/14 [==============================] - 17s 1s/step - loss: 0.1728 - accuracy: 0.9220 - val_loss: 0.0573 - val_accuracy: 0.975
         2
         Epoch 8/10
         14/14 [==============================] - 17s 1s/step - loss: 0.1838 - accuracy: 0.9358 - val_loss: 0.0464 - val_accuracy: 0.991
         7
         Epoch 9/10
         14/14 [==============================] - 17s 1s/step - loss: 0.2046 - accuracy: 0.8968 - val_loss: 0.0491 - val_accuracy: 0.983
         5
         Epoch 10/10
         14/14 [==============================] - 19s 1s/step - loss: 0.1873 - accuracy: 0.9151 - val_loss: 0.0950 - val_accuracy: 0.966
         9

Out[16]: <keras.callbacks.History at 0x29f78ef53a0>
```

10 ADVANTAGES AND DISADVANTAGES

1. ADVANTAGES:

- More dynamic and wider detection as compared to fixed sensors.
- To detect poaching,and monitor comprehensive animal deaths.
- Proposed methods are very convenient and can easily detect.
- Reduction in cost.
- Fast Response.

10.2 DISADVANTAGES:

- Possibility of lack of appropriate animals for special forests.
- Use of batteries create environmental pollution , introducing extra radiation and cadmium to the forest and animals.
- Determining climate conditions , daily temperature differences , seasonal normal temp values are problematic.
- Soil damage which can result in erosion.

11 CONCLUSION

New wireless technologies and new satellite tracking systems can be adapted to increase the efficiency of the system .New sensors can be produced or existing sensors can be improved to increase robustness of the proposed system. A number of investigations can be made regarding animal behavior in case of fire to improve system reliability. The aspects of using wireless sensor networks for forest tree monitoring and alerting using rare event detection with ultra low power consumption. In this prototype, two sensors (mercury sensor & temperature sensor) which work well for the detection of fire and tree theft were selected, and mesh protocol was used for alert routing and event detection Network lifetime and latency estimation for the deployment scenario showed the implementation feasibility of such a monitoring system for deforestation application.

12 FUTURE SCOPE

Development of Micro-Electrical Systems (MEMS), wireless network systems are expected to be widely in use.MEMS are the combination of electrical devices and mechanical structures at an extremely small scale. Many remarches need to be done so as to implemem MEMS in WSN More over, loT is expected to have dramatic impact in Curves in near future. WSNs will be integrated into end innumerable sensor nodes will join the Internet. They will cooperate with other nodes to sense and to monitor the environment. To overcome disadvantages and looking for implementation of Automatic Fire detection.

Eg: Smart driver system.

13 APPENDIX

13.1 GITUHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-123231659447361