PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

(NALAIYA THIRAN)

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

PROJECT REPORT

TEAM ID:PNT2022TMID06174

TEAM MEMBERS	ROLL NO
VIGNESH S	91761914044
ARUNKUMAR M	91761914005
RAGU A	91761914027
VIGNESHKRISHNAKUMAR V	91762014217

ALAGAPPA CHETTIAR GOVERNMENT COLLEGE OF ENGINEERING AND TECHNOLOGY KARAIKUDI-630 001

2022-2023

INDEX

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

INTRODUCTION

Forest fires have become a serious threat to mankind. Besides providing shelter and protection to a large number of living beings, they have been a major source of food, wood, and a great supply of other products. Since ancient times forests have played an important role in social, economic, and religious activities and have enriched human life in a variety of ways both material and psychological. To protect our nature from these rapidly rising forest fires, we need to be cautious enough of every decision we take which could lead to a disastrous end, once and for all. So, for the early detection of forest fires, we propose an image recognition method based on Convolutional Neural Networks (CNN).

The most up to date information on the current fire season in Europe and in the Mediterranean area is provided by the European Forest Fire Information System EFFIS. Each year this institution provides annual report on the forest fires in Europe, the Middle East and North Africa. According to the latest report, which they provided for 2017, the dramatic effects of wildfires have caused damages of over 1.2 million hectares burnt natural lands in the EU and killed 127 people, including fire fighters and civilians. Over 25% of the total burnt area was in the Natural 2000 network, which destroyed much on the efforts of the EU countries to preserve key natural habitats and to save the biodiversity of Europe for the future generations. The same report says that these fires caused estimated losses of around 10 billion euros. Despite these large numbers, EFFIS informs also that the report is showing a decrease in the number of fires, compared to the number of fires, which occurred annually during the last decade. This decrease can be explained with the more severe actions and sanctions to the people that caused the wildfires and with the introduction of more advanced technical solutions for early detection of fires. Obviously, the fight against fires can mitigate the damages, but the numbers, which represent the burnt area and the human lives, are still huge.

The most important factors in the fight against forest fires include the earliest possible detection of the fire event, the proper categorization of the fire and fast response from the firefighting departments. The aim of the proposed platform is not only to use modern technologies, but also to improve the abovementioned factors by reducing the fire detection time, by minimizing the false alarms and by issuing of timely responses and notifications to the fire services in case of real forest fires.

1.1 PROJECT OVERVIEW:

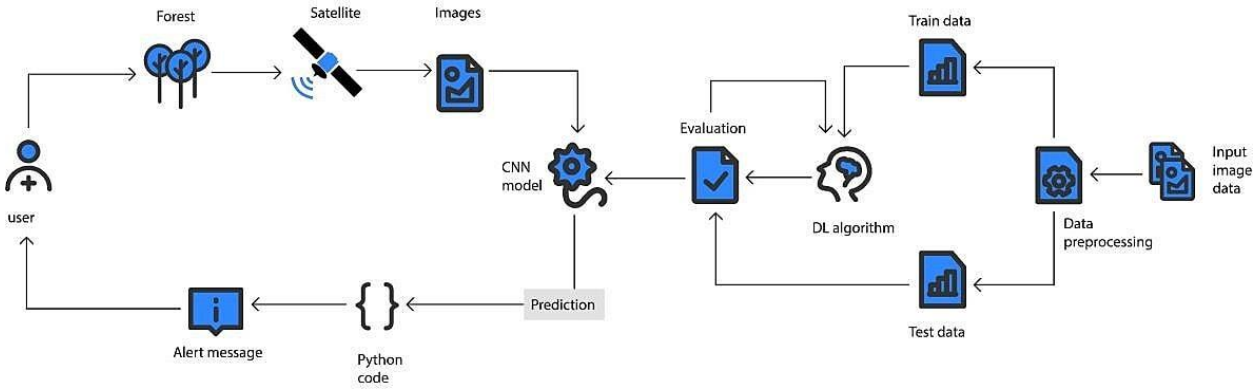
Project Objective:

➤ 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. In this paper, we discuss a new approach for fire detection and control, in which modern technologies are used.

➤ With the help of Artificial Intelligence (AI) using computer vision methods for recognition and detection of smoke or fire, based on the still images or the video input from the Satellites. Using ML, the system will be able to calculate the percentage of damage caused, if any.

➤ This project primarily works into early detection & prevention. Nevertheless, the issue of forest fire control provides a worthy space and exciting area for future research.



1.2 Purpose:

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.

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.

2.Literature Survey:

2.1 Existing problem:

Forest fires are a matter of concern because they cause extensive damage to environment, property and human life. Hence, it is crucial to detect the forest fire at an earlier stage. This can help in saving flora and fauna of the region along with the resources. Also, it may help to control the spread of fire at initial phase. The task of monitoring the forests is difficult because of the vast territory and dense forest. The wide ranging adverse ecological, economic and social impacts of forest fires including forest degradation are:

- loss of valuable wood resources
- deterioration of catchment areas

The forest fire has become a threat to not only to the forest wealth but also flora and fauna and ecology of the environment of the region. The main cause of forest fires can be categorized under natural and man-made classes. High atmospheric temperature, lightening and dryness offer positive environment for a fire to start which are the natural causes for forest fire. The fire is also caused by Man-made sources like naked flame, cigarette, electric spark etc **...**

2.2 References:

Title	Authors	Techniques Used	Merits	Demerits	PublishedDate
Image Processing Based Forest Fire Detection	Vipin V	Rule based color model for forest fire pixel classification	Fire detectionrate is higher	False alarm rate is higher	February2012
A Literature Study on Image Processing for Forest Fire Detection	Priyadarshini M Hanamaraddi	Conversion of RGB to YCbCr image	Complexity of the system is very less	False alarmrate is high	January2016
Forest Fire Detection usinga Rule-Based Image Processing Algorithm and Temporal Variation	Mubarak A.I. Mahmoud and Honge Ren	Fire Detection Algorithm which combines the saturation channel of the HSV color and the RGB color	Better in separating the luminance from the chrominance and has good detection rate	System needs to be improvedby using a combination ofrules	May,2018
Image Processing Based Forest Fire DetectionUsing InfraredCamera	Norsuzilla Ya'acob,Mohammed Syamirza Mohd Najib,Noraisyah Tajudin,Azita	Mapping and tracking of forest fire using cameras	Simple and less expensive technique	Difficult to differentiate object with same color as fire	January,2021

	Laily Yusof and Murizah Kassim				
Forest Fire Detection and Identification Using Image Processing and SVM	Mubarak Adam Ishag Mahmoud	Multi stages of image processing and support vector machine (SVM)	Good detection rate and low false alarm	Experimentedonly in simulation	July, 2020

Forest Fire Detection using Combined Architecture of Separable Convolution and Image Processing	Navyashree J,VeenaM Naik	CNN model with regularizationand digital image processing	Acceptable performance	Computationallycostly implementation of CNN architecture	September, 2022
Forest Fire Prediction using Image Processing and Machine Learning	Mohana kumar S,Sowmya B,Priyanka S,Ruchita Sharma,ShivankTej Spoorthi Ashok karani	Image processing andR-CNN model	Reliable and decent accuracy	Some improvementsare needed to increase speedand accuracy	March,2021

-

2.3 Problem Statement Definition:

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

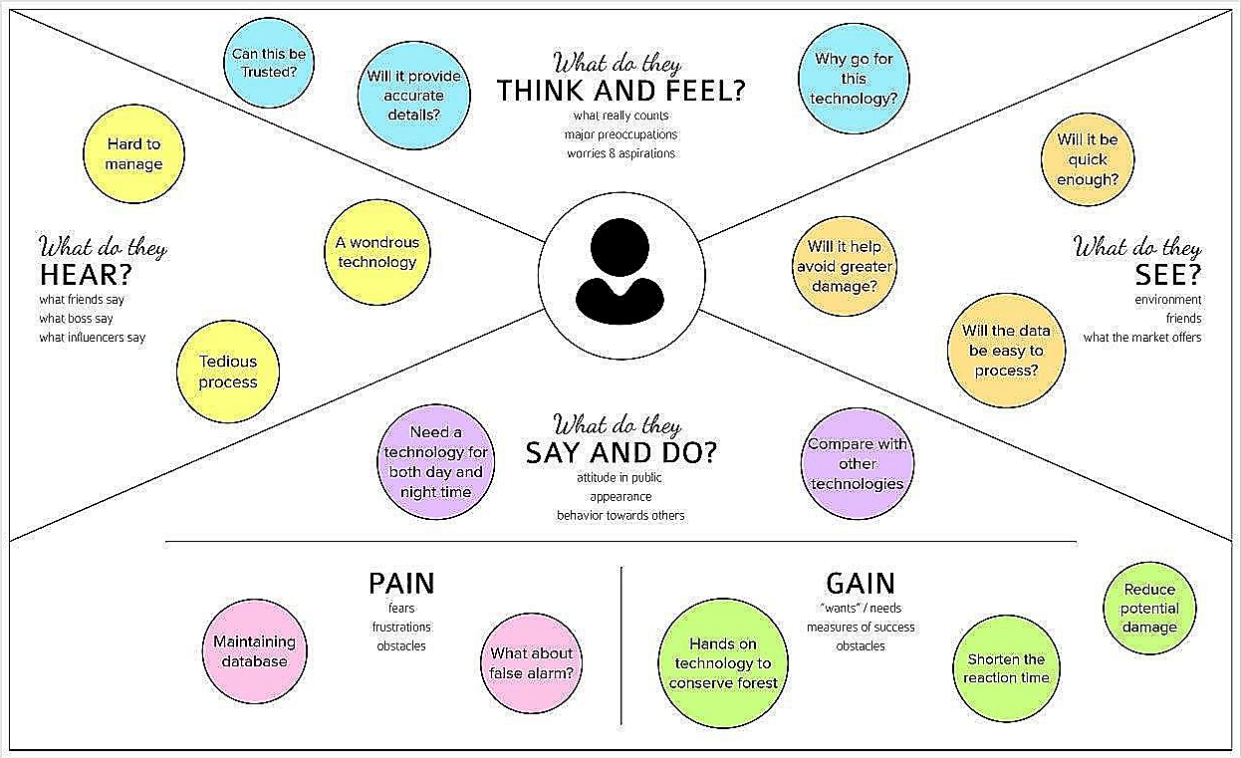
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It helps the team to understand their users in a better way.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

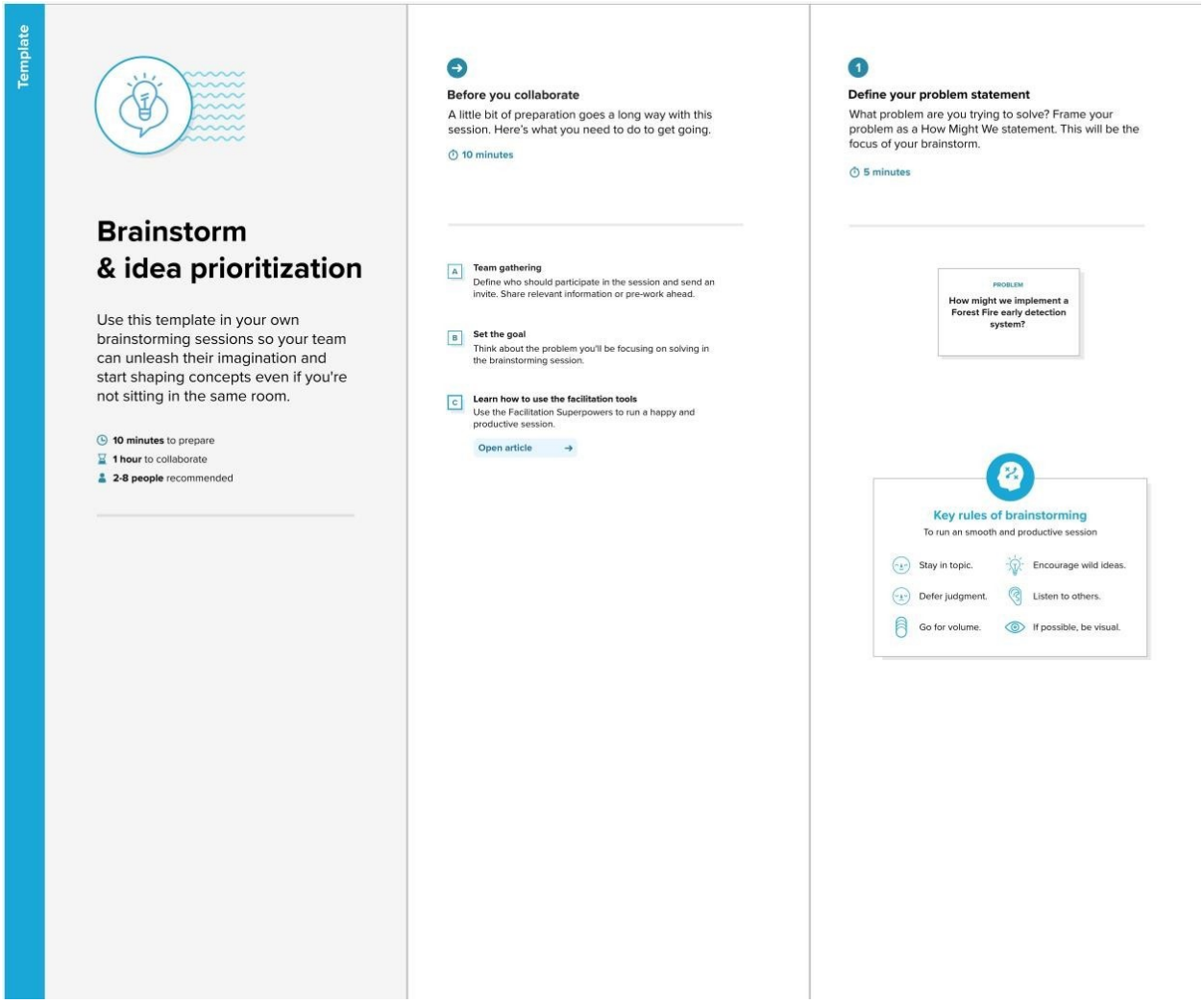


3.2 Brainstorm & Idea Prioritization Template:

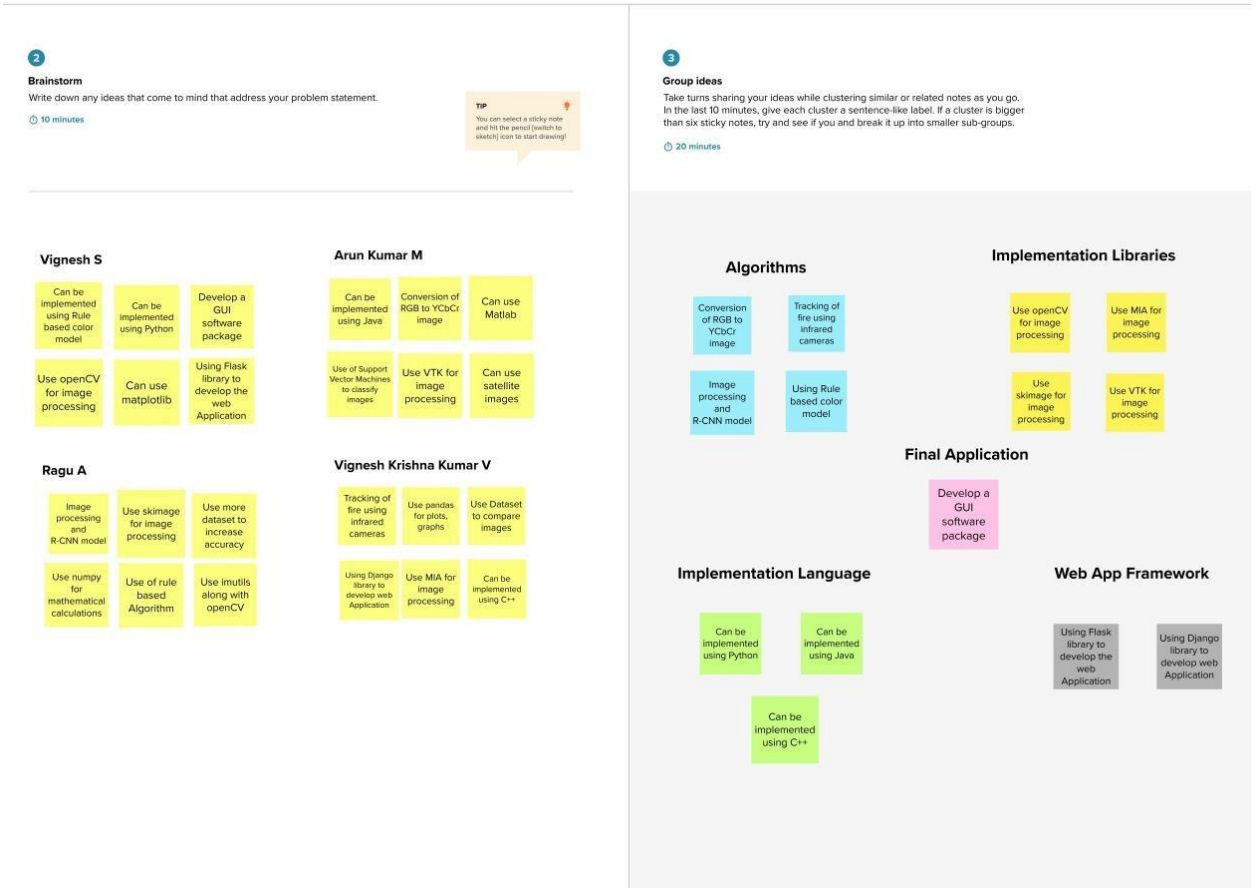
Brainstorming provides a free and open environment that encourages everyone within a teamto participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants areencouraged to collaborate, helping each other develop a rich amount of creative solutions.

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

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping:



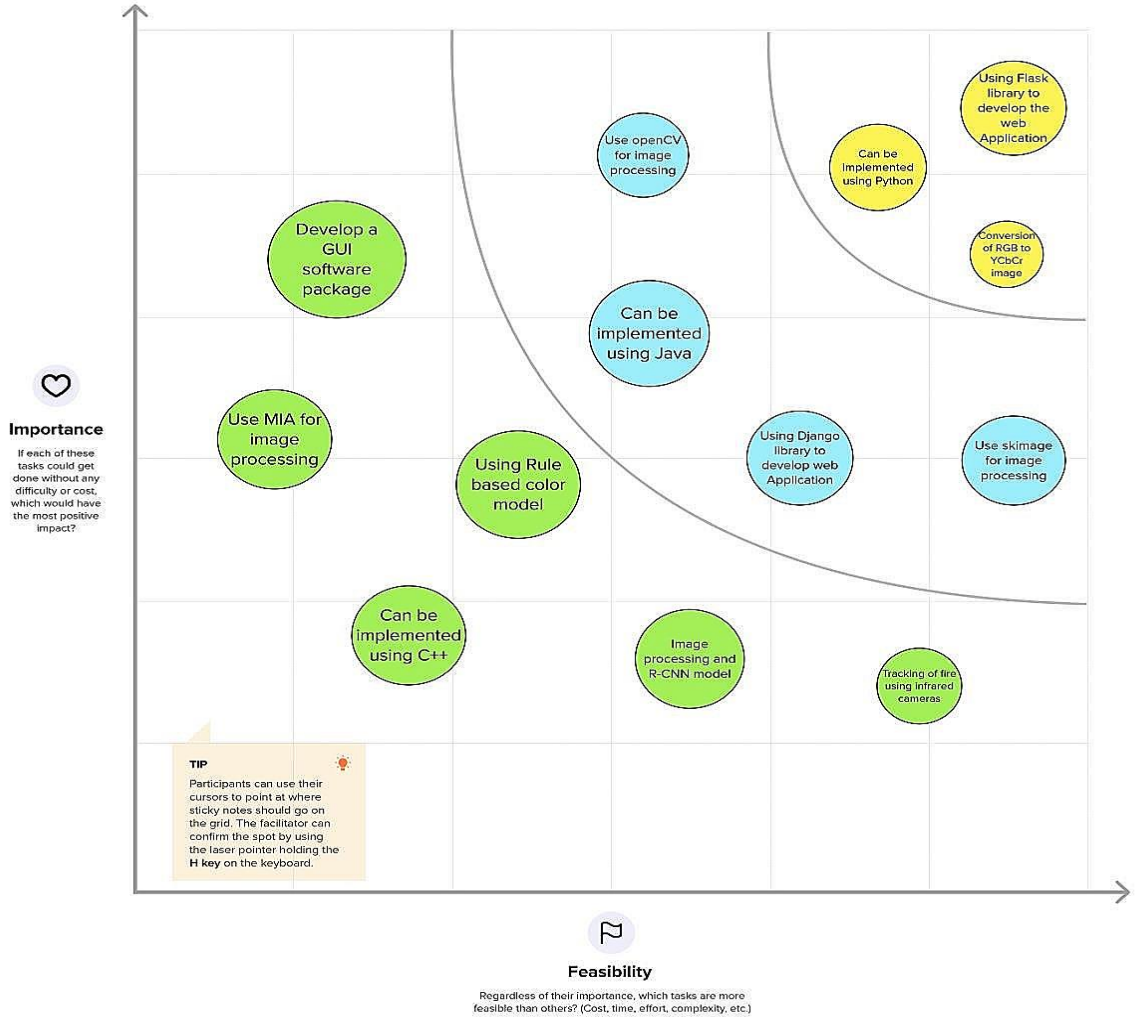
Step-3: Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3.3 Proposed Solution:

Project team shall fill the following information in proposed solution template.

S.No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Human monitoring to detect forest fires requires 24X7 monitoring. Also involves delay in transferring information.
2.	Idea / Solution description	To avoid delay and manual monitoring, we canuse image processing to monitor forests.
3.	Novelty / Uniqueness	Satellite images of forests will be processed foridentifying fire.
4.	Social Impact/ Customer Satisfaction	It will help prevent loss of natural resources and wildlife.
5.	Business Model (Revenue Model)	It can reduce the reaction time and accuracy ishigh.
6.	Scalability of the Solution	In future we can include machine learning todetect smoke coming from forest.

3.4 Problem Solution fit:

4.Requirement Analysis

4.1 Functional requirement:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	Dataset Collection	A huge dataset is needed to train the model. Images offorest fire needs to be processed to train the model.
FR-2	Image Pre-Processing	Image pre-processing involves applying Image DataGenerator on train set and test set.
FR-3	Model Building	A part of the collected dataset will be used to build model. Building model involves adding CNN layers, hidden layers and output layers, configure the learningprocess and optimize the model.
FR-4	Model Training	The other part of the dataset will be used to train themodel. Training will be done a number of times to optimize the process.
FR-5	Setting Software	After multiple testing and training, the model built willbe used for setting the software.

4.2 Non-functional Requirements:

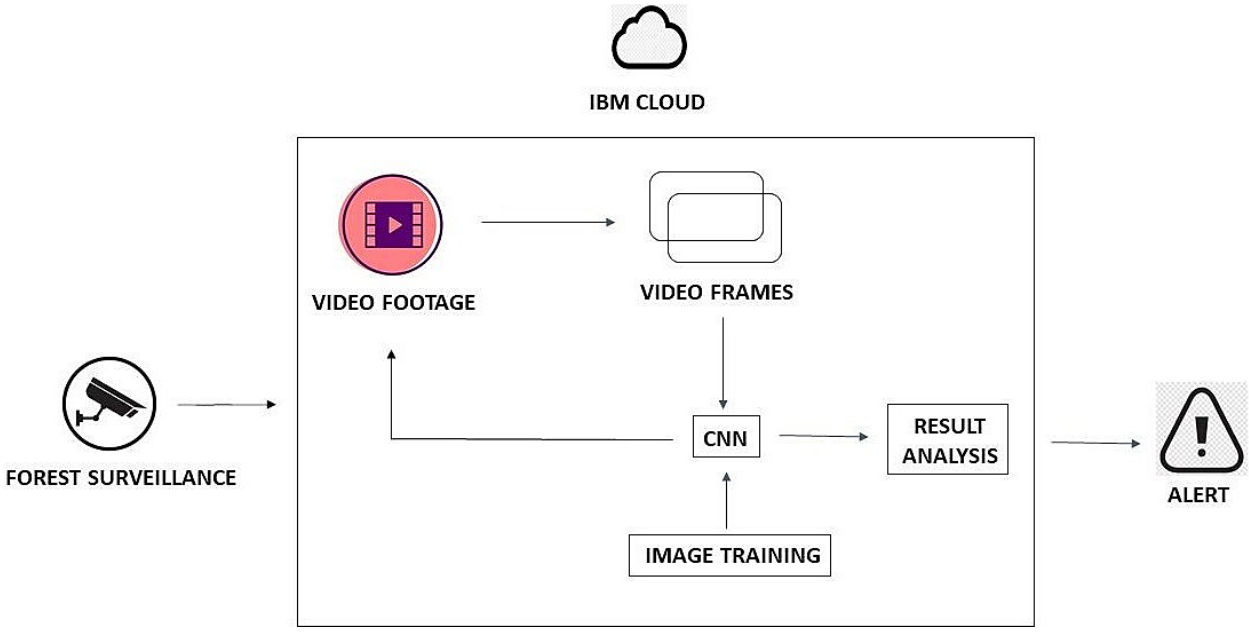
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Forest fire detection is one of the practically important issues in our world. The most difficult taskis to detect fire and alert without any delay. This software will be efficient in detecting and alerting.
NFR-2	**Security**	Using python Flask in connecting cloud will providesecurity to the project.
NFR-3	**Reliability**	This system not only increases the sensing time offire, but also reduce complexities.
NFR-4	**Performance**	By increasing the number of training examples, theaccuracy of the system can be increased. The performance of this method will be significantly better than all conventional methods.
NFR-5	**Availability**	Since we use cloud, the availability of this softwareis all over the world only Internet facility is needed.
NFR-6	**Scalability**	This technique could be used for analyzing soilconditions and water pollution in future.

5.Project Design:

5.1 Data Flow Diagrams:

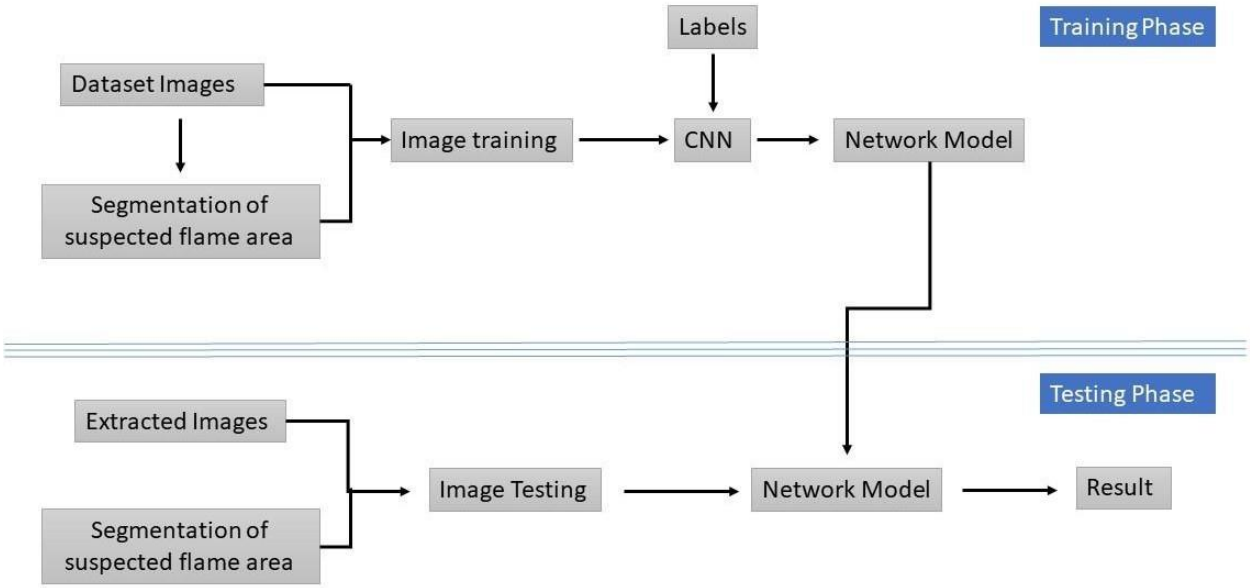
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. Aneat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



Data Flow Diagram (Simplified)

1. Getting forest surveillance footage.
2. Converting the video footage into frames.
3. The converted frames are given to CNN model.
4. CNN model predicts the image frames and gives to the python program.
5. The action is performed on the image according to the prediction from the model by the python program.
6. The resultant image can be viewed through the application web UI.

Data Flow Diagram (DFD)

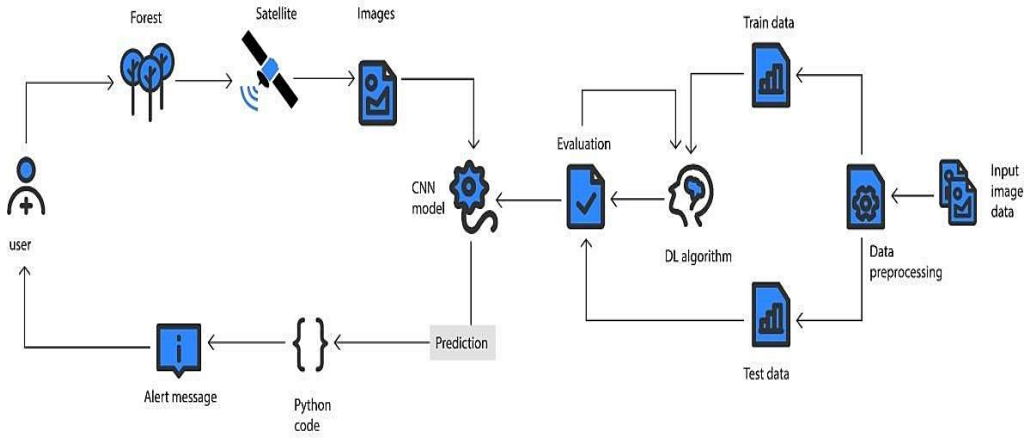


5.2 Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridgesthe gap between business problems and technology solutions. Its goals are to:

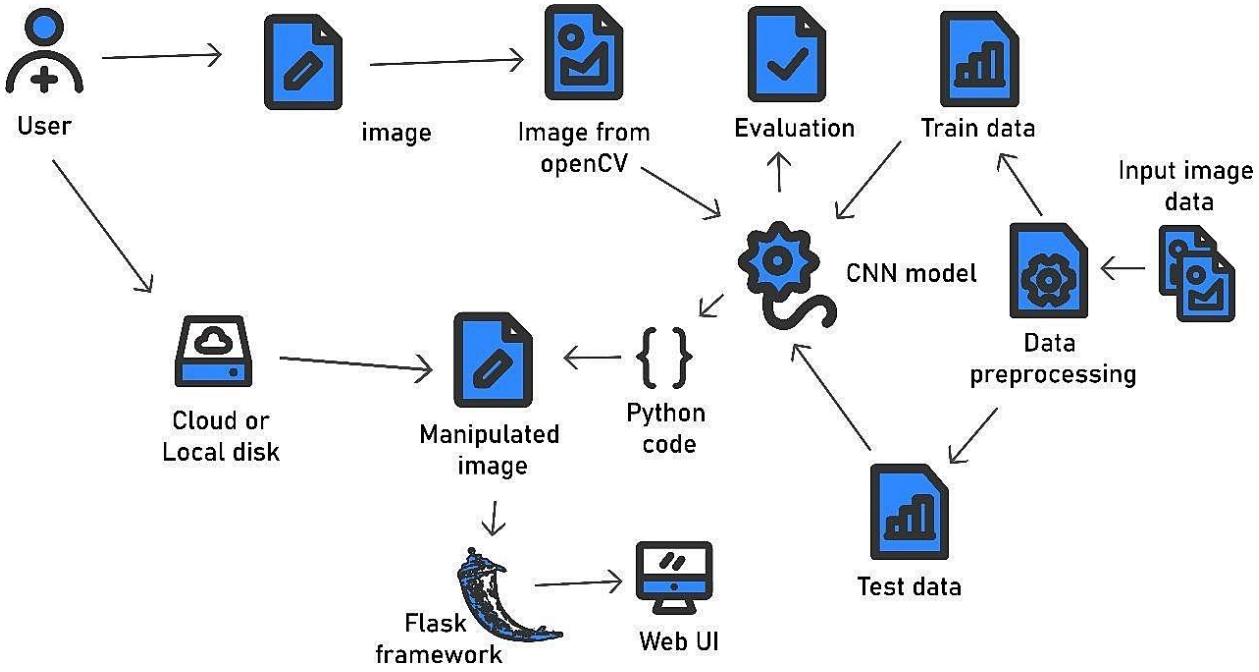
1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behavior, and other aspects of thesoftware to project stakeholders.
3. Define features, development phases, and solution requirements.
4. Provide specifications according to which the solution is defined, managed,and delivered.

Emerging Methods for Early Detection of Forest Fires



Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 &table 2



5.3 User stories

Use the below template to list all the user stories for the product.

User Type	FunctionalRequirement (Epic)	User Story Number	User Story/Task	Acceptance criteria	Priority
Forest Officer	Surveillance	USN-1	As a user, I will get the live video footages of forest area for monitoring	Uploading multiple forest images (.jpg, .png, etc.,)	High
	Capturing liveforest	USN-2	Real time forest Image frames are extracted from video that capturedvia satellite or drone.	Different forest fire inputs to perform specific action on images.	High
	Image processing	USN-3	The input images are processed to detect fire spot areas.	Training the model for better accuracy	High
	Prediction	USN-4	The input frames are predicted based on the developed CNN model	Better accuracy is obtained only by repeated training and testing	Medium

	Performaction	USN-5	Performing an action on images	Crop, rotate, zoom actions are performed.	High
	Viewing result	USN-6	Viewing the resultant processed image.	The resultant imageis displayed in the web UI.	High

6.Project Planning &Scheduling:

6.1 Sprint Planning &Estimation:

Project Planning Phase
Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collecting Dataset	USN-1	To create a model with great accuracy, we needto collect a huge dataset.	10	High	All Members
Sprint-1	Data Pre-processing	USN-2	The collected data can be uneven in size and indifferent formats. So, data pre-processing is required	10	Medium	All Members
Sprint-1	Configuring required libraries	USN-3	Import and configure the ImageDataGeneratorLibrary and class	10	High	All Members

Sprint-1	Image Augmentation	USN-4	As a user, I can register for the application through Facebook	10	Medium	All Members
Sprint-2	Model Building	USN-5	Importing model building libraries and initializing the model	10	High	All Members
Sprint-2		USN-6	Adding CNN layers and Dense layers	10	High	All Members
Sprint-2		USN-7	A part of the collected dataset is fed to the model for training and is repeated until we get abetter accuracy	10	High	All Members

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-3	Application Building	USN-8	Create Web Application using HTML, CSS, JavaScript	10	Medium	All Members
Sprint-3		USN-9	Build Python code	10	High	All Members
Sprint-3		USN-10	Run the Application	10	High	All Members

Sprint-4	Train the model on IBM	USN-11	Register for IBM Cloud	10	High	All Members
Sprint-4		USN-12	Train the Model and test the Model and itsOverall Performance	10	High	All Members

6.2 Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart:

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov2022	10	05 Nov2022
Sprint-3	10	6 Days	07 Nov2022	12 Nov2022	10	12 Nov2022
Sprint-4	10	6 Days	14 Nov2022	19 Nov2022	10	19 Nov2022

Velocity:

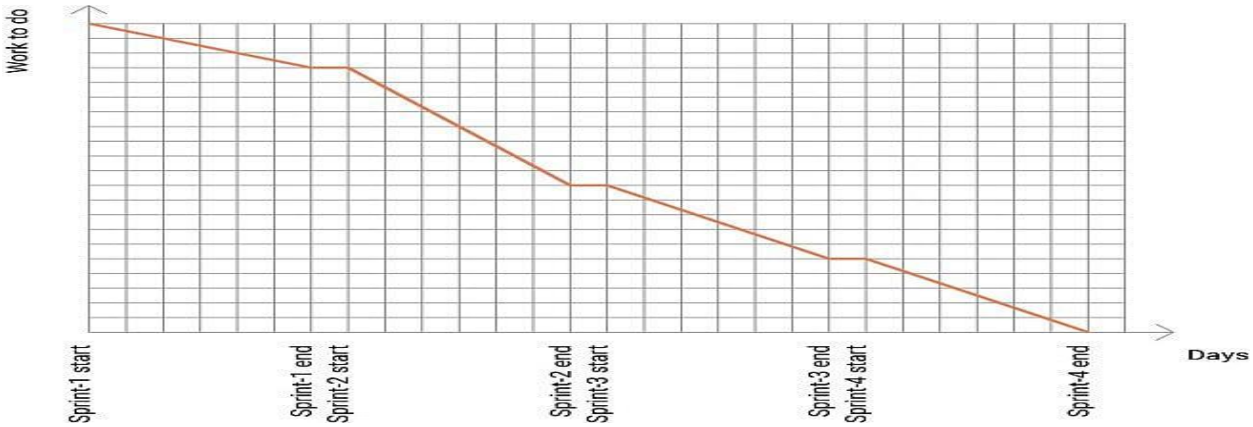
Imagine we have a 10-days print duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = 10/6 = 1.67$$

6.3 Reports from JIRA:

Burndown Chart:

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



7.Coding and Solutioning:

7.1 Feature 1:

NUMPY :

When you call the statement import numpy as np, you are shortening the phrase "numpy" to "np" to make your code easier to read. It also helps to avoid namespace issues. NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

KERAS:

Keras is an open source Neural Network library written in Python that runs on top of Theano or TensorFlow. It is designed to be modular, fast and easy to use. Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend.
Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano. Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras in Python doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

MATPLOTLIB.PYPLOT

Pyplot is a collection of functions in the popular visualization package Matplotlib. Its functions manipulate elements of a figure, such as creating a figure, creating a plotting area, plotting lines, adding plot labels, etc. Let's use the plot() function from **pyplot** to create a dashed line graph showing the growth of a company's stock. Remember, you can change the color of the line by adding the argument color and the line style by adding the argument line style.
Two lists, days (representing the days since the company became public), and prices (representing the price of the stock corresponding to that day) are available in your workspace.

TENSORFLOW:

TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses data flow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources.
Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.
TensorFlow enables you to build data flow graphs and structures to define how data moves through a graph by taking inputs as a multi-dimensional array called Tensor. It allows you to construct a flowchart of operations that can be performed on these inputs, which goes at one end and comes at the other end as output.

Playsound :

pip install playsound

The playsound module contains only a single function named playsound().
It requires one argument: the path to the file with the sound we have to play. It can be a *local file*, or a *URL*. There's an optional second argument**,** block, which is set to *True* by default. We can set it to *False* for making the function run asynchronously.
It works with both WAV and MP3 files.

Example: For WAV format

ImageDataGenerator:

Keras ImageDataGenerator to perform image augmentation by directly reading the CSV files through pandas data frame.

Most often the Image datasets available on the internet are either has images placed under folders which has their respective class names or placed under a single folder along with a CSV or JSON file which maps the image filenames with their corresponding classes.

In former case, we already have the flow from directory method that helps you read the images from the folders, but in the later case you will need to write either a custom generator or move the image files to their respective class name folders and use flow from directory to utilize ImageDataGenerator, but now with the help of flow from data frame method you can directly pass the Pandas Data Frame which has the mapping between filenames of the images and their labels.

```
def startPredict():
  alarm = False
  while (1):
    success, frame = video.read()
    if (success == False):
      break
    img1 = cv2.resize(frame, (64, 64))
    y = np.array(img1)
    x = np.expand_dims(y, axis=0)
    pred = model.predict(x)
    list1 = pred.tolist()
    p = int(list1[0][0])
    if (p == 1):
      if not alarm:
        alarm = True
        t2.start()
         print(message.sid)
        print('Fire Detected')
    else:
      print('No Fire')
```

7.2 Feature 2

```
def startPredict():
    alarm = False
    while (1):
        success, frame = video.read()
        if (success == False):
            break
        img1 = cv2.resize(frame, (64, 64))
        y = np.array(img1)
        x = np.expand_dims(y, axis=0)
        pred = model.predict(x)
        list1 = pred.tolist()
        p = int(list1[0][0])
        if (p == 1):
            if not alarm:
                alarm = True
                account_sid = 'ACdfd0daf5d349b93e110c14acdbe25644'
                auth_token = 'e19cf3b8e0bc79bedb71f27cc02a94a3'
                client = Client(account_sid, auth_token)
                message = client.messages.create(
                    body='Forest fire Detected, stay alert',
                    from_='+18088099455',
                    to='+919894764684')
                t2.start()
                print(message.sid)
                print('Fire Detected')
                print('SMS sent')
        else:
            print('No Fire')
```

8.TESTING:

8.1 Test Case:

S.No	Feature type	Component	Test scenario	Steps to execute	Expected Result	Actual Result	Status
1	Functional	Forest fire video	Check whether the model detects fire	Check whether the uploaded video contains fire	Fire Detected	Fire Detected	Pass
2	Functional	Forest video without fire	Check whether the model detects correctly	Check whether the model detects correctly	No Fire	No Fire	Pass
3	Functional	Forest video with smoke	Check whether the model detects smoke correctly as fire	Check whether the uploaded video contains fire	Fire Detected	Fire Detected	Pass

Purpose of document :

The purpose of this document is to briefly explain the test coverage and open issues of the project - Emerging Methods for Early Detection of Forest Fires at the time of the release to User Acceptance Testing (UAT).

Defect analysis :

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

Stage	Severity 1(High)	Severity 2	Severity 3	Severity 4(Low)	Subtotal
Design	0	1	0	2	3
Coding and solutioning	2	1	2	4	9
Executing	1	1	0	1	3
Total	3	3	2	7	15
Fixed bugs	3	3	2	7	15
Not fixed	0	0	0	0	0

Test Case Analysis :

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

Section	Total test cases	Not tested	Fail	Pass
Preprocessing the images from video	1	0	0	1
Prediction	2	0	0	2
Alarming when fire detected	1	0	0	1
Sending alert message	1	0	0	1

9.RESULT:

Performance Metrics:

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No	Parameter	Values	Screenshot
1.	Model Summary	Total params: 3,937,538 Trainable params: 3,937,538 Non-trainable params: 0	`1 model.summary()` ✓ 0.9s Python Model: "sequential_1" _____ Layer (type) Output Shape Param # ===== conv2d_1 (Conv2D) (None, 62, 62, 32) 896 max_pooling2d_1 (MaxPooling2D) (None, 31, 31, 32) 0 flatten_1 (Flatten) (None, 30752) 0 dense_1 (Dense) (None, 128) 3936384 dense_2 (Dense) (None, 2) 258 ===== Total params: 3,937,538 Trainable params: 3,937,538 Non-trainable params: 0 _____

2.	Accuracy	Training Accuracy - 0.9754
		
		Validation Accuracy – 0.9265
			`1 # Training the model`
			`2 r=model.fit(train_dataset,epochs=8,validation_data=test_dataset)`
			Python
			Epoch 1/8
			58/58 [==========] - 89s 2s/step - loss: 0.3045 - accuracy: 0.8788 - val_loss: 0.3466 - val_accuracy: 0.8676
			Epoch 2/8
			58/58 [==========] - 87s 1s/step - loss: 0.1550 - accuracy: 0.9536 - val_loss: 0.2345 - val_accuracy: 0.9118
			Epoch 3/8
			58/58 [==========] - 87s 2s/step - loss: 0.1368 - accuracy: 0.9525 - val_loss: 0.2384 - val_accuracy: 0.8971
			Epoch 4/8
			58/58 [==========] - 87s 1s/step - loss: 0.1161 - accuracy: 0.9634 - val_loss: 0.6027 - val_accuracy: 0.8529
			Epoch 5/8
			58/58 [==========] - 87s 1s/step - loss: 0.1168 - accuracy: 0.9645 - val_loss: 0.2641 - val_accuracy: 0.9118
			Epoch 6/8
			58/58 [==========] - 87s 1s/step - loss: 0.1005 - accuracy: 0.9705 - val_loss: 0.1197 - val_accuracy: 0.9559
			Epoch 7/8
			58/58 [==========] - 88s 2s/step - loss: 0.0762 - accuracy: 0.9727 - val_loss: 0.1655 - val_accuracy: 0.9118
			Epoch 8/8
			58/58 [==========] - 85s 1s/step - loss: 0.0695 - accuracy: 0.9754 - val_loss: 0.1816 - val_accuracy: 0.9265

10.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

- ✓ Easy to detect forest fires.
- ✓ Highly efficient.
- ✓ Alert message is directly sent to the respective people.
- ✓ To find the fires in rough areas.
- ✓ Disaster can be prevented before it become serious.

DISADVANTAGES:

- Signal loss.
- It is not 100% accurate.
- Alert signal may delay in fraction of second

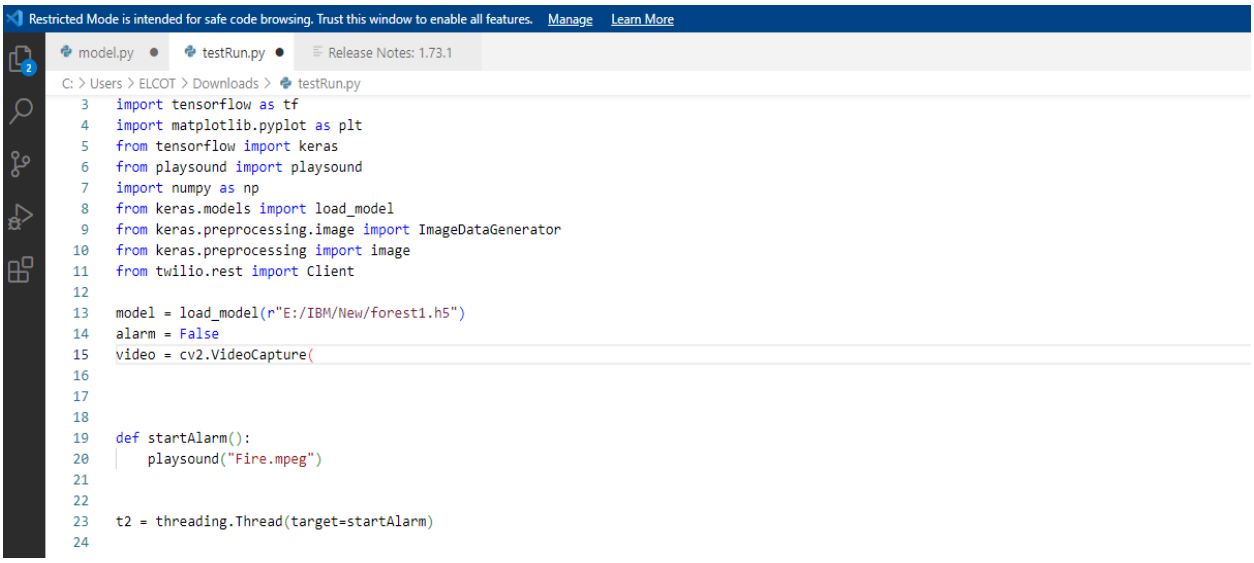
11.Conclusion:

The system for early forest fire detection is still in its development stage. We are still waiting for some equipment to be purchased, but we have planned and discussed the actual implementation. We have performed a thorough research and some simulation experiments and we believe that we follow the right way to achieve the goal. We also believe that we apply adequate approach that is also up-to-date. We think that the system could enhance the available platforms for fire detection and we hope that such improvement could significantly reduce the damages caused by untimely or late fire detection.

12.Future Scope:

The application can be enhanced by training the model with a larger dataset consisting of fires at various stages and dimensions. With higher GPU memory, we could use two deep learning models for feature extraction, whose output feature vectors are concatenated and classified to offer more robustness. An R-CNN model can be used to implement fire localization along with classification. We can also expect better deep learning architectures to emerge in the future, offering better feature extraction. The application will also offer a considerably better performance when run on.

13.Appendix:

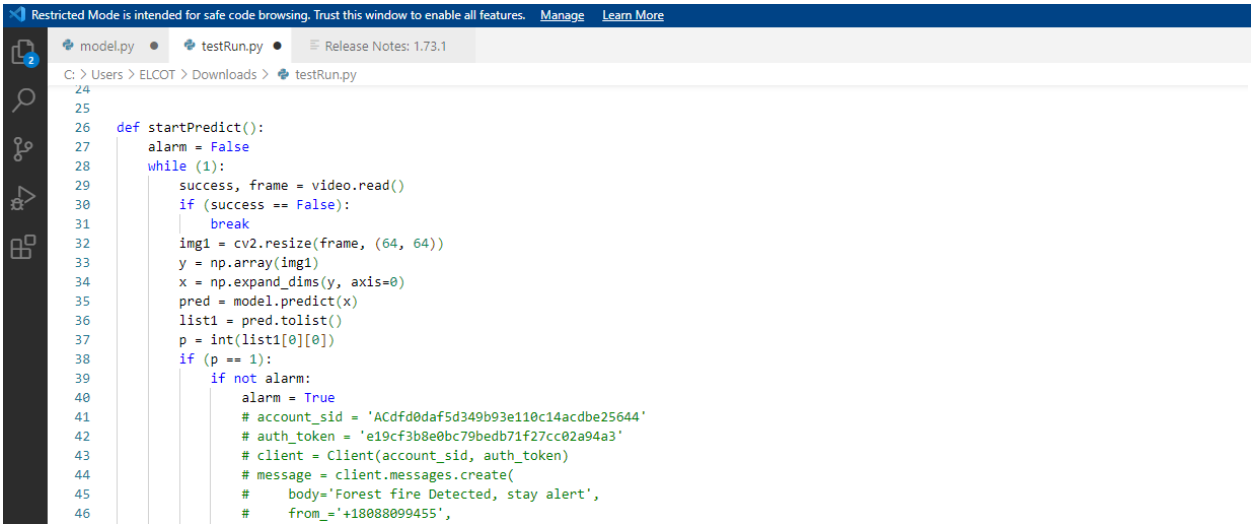


Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

model.py testRun.py Release Notes: 1.73.1

C: > Users > ELCOT > Downloads > testRun.py

```
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow import keras
from playsound import playsound
import numpy as np
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from twilio.rest import Client

model = load_model(r"E:/IBM/New/forest1.h5")
alarm = False
video = cv2.VideoCapture(



def startAlarm():
    playsound("Fire.mpeg")


t2 = threading.Thread(target=startAlarm)

```

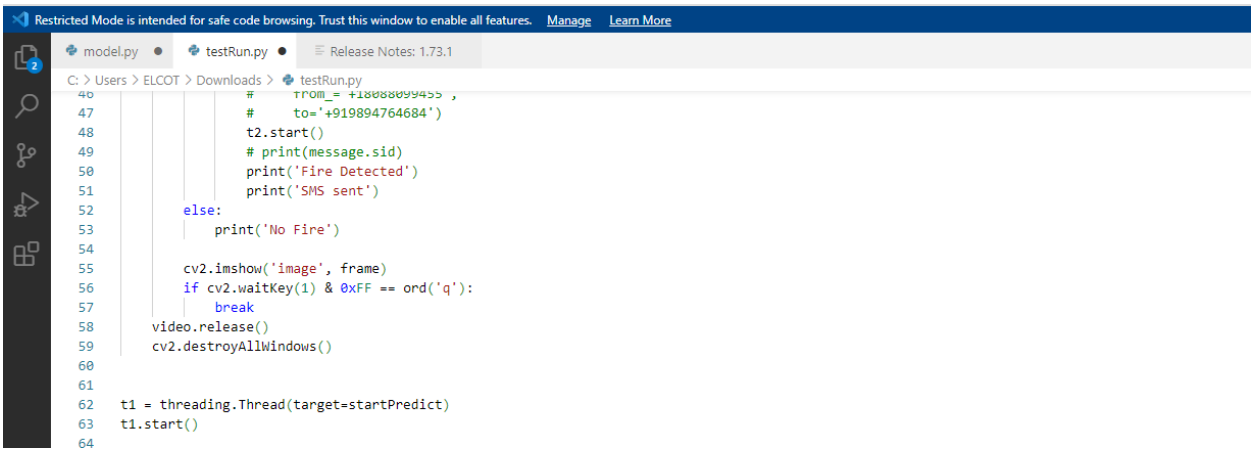


Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

model.py testRun.py Release Notes: 1.73.1

C: > Users > ELCOT > Downloads > testRun.py

```


def startPredict():
    alarm = False
    while (1):
        success, frame = video.read()
        if (success == False):
            break
        img1 = cv2.resize(frame, (64, 64))
        y = np.array(img1)
        x = np.expand_dims(y, axis=0)
        pred = model.predict(x)
        list1 = pred.tolist()
        p = int(list1[0][0])
        if (p == 1):
            if not alarm:
                alarm = True
                # account_sid = 'ACdfd0daf5d349b93e110c14acdbe25644'
                # auth_token = 'e19cf3b8e0bc79bedb71f27cc02a94a3'
                # client = Client(account_sid, auth_token)
                # message = client.messages.create(
                #     body='Forest fire Detected, stay alert',
                #     from_='+18088099455',
```



TEST CODE:

```
import threading
import cv2
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow import keras
from playsound import playsound
import numpy as np
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from twilio.rest import Client

model = load_model(r"E:/IBM/New/forest1.h5")
alarm = False
video = cv2.VideoCapture(



def startAlarm():
   playsound("Fire.mpeg")


t2 = threading.Thread(target=startAlarm)
```

```
def startPredict():
    alarm = False
    while (1):
        success, frame = video.read()
        if (success == False):
            break
        img1 = cv2.resize(frame, (64, 64))
        y = np.array(img1)
        x = np.expand_dims(y, axis=0)
        pred = model.predict(x)
        list1 = pred.tolist()
        p = int(list1[0][0])
        if (p == 1):
            if not alarm:
                alarm = True
                account_sid = 'ACdfd0daf5d349b93e110c14acdbe25644'
                auth_token = 'e19cf3b8e0bc79bedb71f27cc02a94a3'
                client = Client(account_sid, auth_token)
                message = client.messages.create(
                    body='Forest fire Detected, stay alert',
                    from_='+18088099455',
                    to='+919894764684')
                t2.start()
                print(message.sid)
                print('Fire Detected')
                print('SMS sent')
        else:
            print('No Fire')

        cv2.imshow('image', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    video.release()
    cv2.destroyAllWindows()


t1 = threading.Thread(target=startPredict)
t1.start()
```

13. GitHub & Project Demo Link:

GitHub Repository Link: https://github.com/IBM-EPBL/IBM-Project-16819-1659623561

Project Demo Link: https://www.youtube.com/embed/uXuH8ExBXGM