A PROJECT REPORT ON

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

Domain: ARTIFICIAL INTELLIGENCE

Team ID: PNT2022TMID51669

College Name: Mar Ephraem College of Engineering and Technology

S. SHERJIN RAJ
(961419104041)
Department of Computer Science Engineering

R.M. ANTO SHINE JOVEMON
(961419104013)
Department of Computer Science Engineering

J. ADARSH
(961419104004)
Department of Computer Science Engineering

P.U. SHABIN
(961419104039)
Department of Computer Science Engineering

J. DESMOND SAMUEL
(961419104023)
Department of Computer Science Engineering

TABLE OF CONTENTS

1.INTRODUCTION

1.1 Project Overview

Forests, which are diverse centers of flora and wildlife and create 1/3 of the world's oxygen, are at risk of forest fires, both natural and man-made. The precaution of averting sucha massive devastating flare can save many animals and the environment. Protecting forests before they are harmed is a method of repaying Mother Nature's everlasting gift.

Wildfires are one of the biggest catastrophes faced by our society today causing irrevocable damages.These forest fires can be man-made or caused by mothernature by different weather conditions, torrential winds. These fires cause damages not only to the environment they also destroy vast homes and property.

1.2 Purpose

Forest fires have become a major threat around the world, causing many negative impacts on human habitats and forest ecosystems. Climatic changes and the greenhouse effect are some of the consequences of such destruction. A higher percentage of forestfires occur due to human activities.Thegoal of the project is to develop a forest fire detection system that can identify forest fires intheir early phases.

2. LITERATURE SURVEY

2.1 Existing Problem

Every year, there are an estimated340,000 premature deaths from respiratory and cardiovascular issues attributed to wildfire smoke.

The increasing frequency and severity of wildfires pose a growingthreat to biodiversity globally. Individuals, companies and public authorities bear great economic costs due to fires. In order to reduce all these, we need to detect the forest fire at an early stage and prevent it.

2.2 References

- ***Turgay Celik, Huseyin Ozkaramanl, and Hassan Demirel (2007). Fire and Smoke detection without Sensors: Image Processing based approach.15th European signal processing conference (eusipco 2007), Poznan, Poland, September 3-7.***
- ***Osman Gunay, A. Enis C, Etin, Yusuf Hakan, Habiboglu. Flame Detection method in video using Covariance descriptors, IEEE transactions, 1817-1820.***
- ***CHENG Caixia, SUN Fuchun, ZHOU Xinquan (2011).***

One Fire Detection Method Using Neural Networks, Tsinghua Science and Technology, ISSN 1007-0214 05/17 31-35Volume 16, Number 1.

- ***S. A. Christopher, M. Wang, T. A. Berendes, and R. M. Welch (1998). The 1985 biomass burning season in South America: Satellite remote sensing of fires, smoke, and regional radiative energy budgets, vol. 37, 661– 678***

- ***Paulo Vinicius Koerich Borges (2010). A Probabilistic Approach for VisionBased Fire Detection in Videos, IEEE transactions on circuits and systems for video technology, vol. 20, no. 5.***

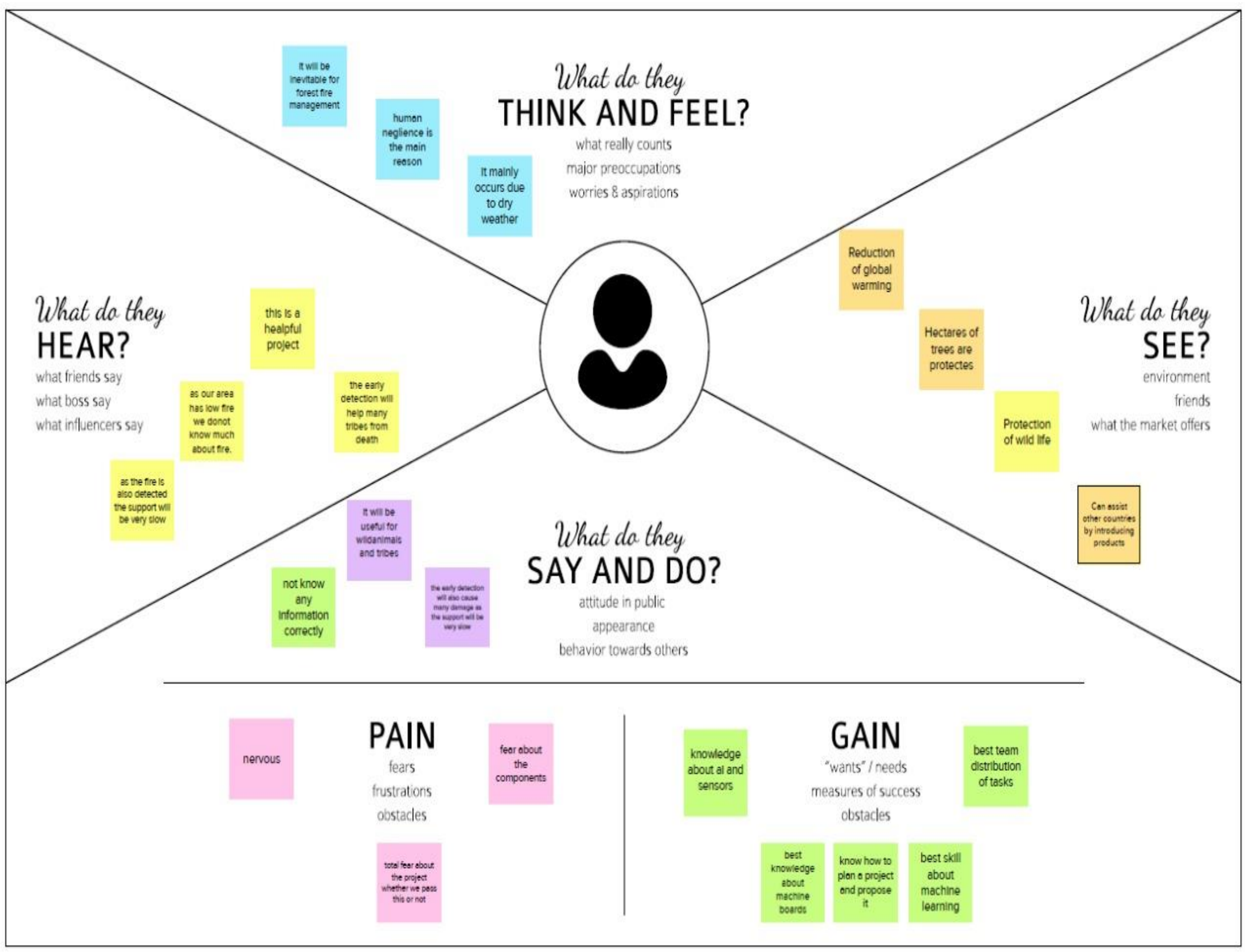
- ***Jiawei Han, Micheline Kamber, Jian Pei (2012). Data Mining Concepts and Techniques, Third edition, 248-253, 350-351.***

2.3 Problem Statement Definition

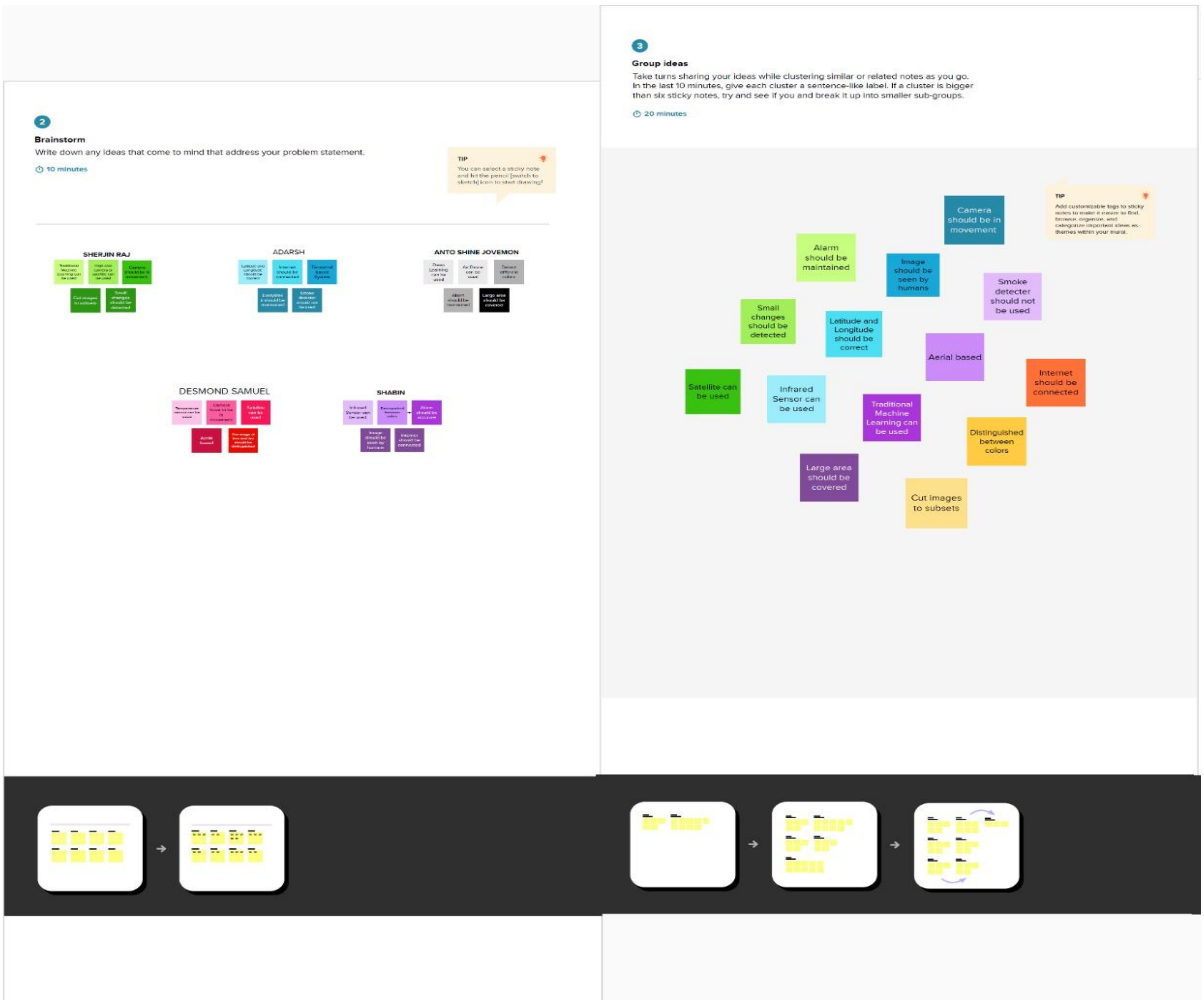
A Large destructive fire that spread over a forest or area of woodland which leads to damage in Wildlife, humans, property and Environment. The major Causes Are Lightning. Sparks from Rock falls. Volcanic Eruption or any other manual Ignition from the Humans on purpose which leads to the following disadvantages: A forest fire sets up the potential for soil erosion to occur, Forest fires always bring death to life of humans and animals, Uncontrolled fires can cause localized air pollution, Homes can be destroyed without compensation

3. IDEATION AND PROPOSED SOLUTION

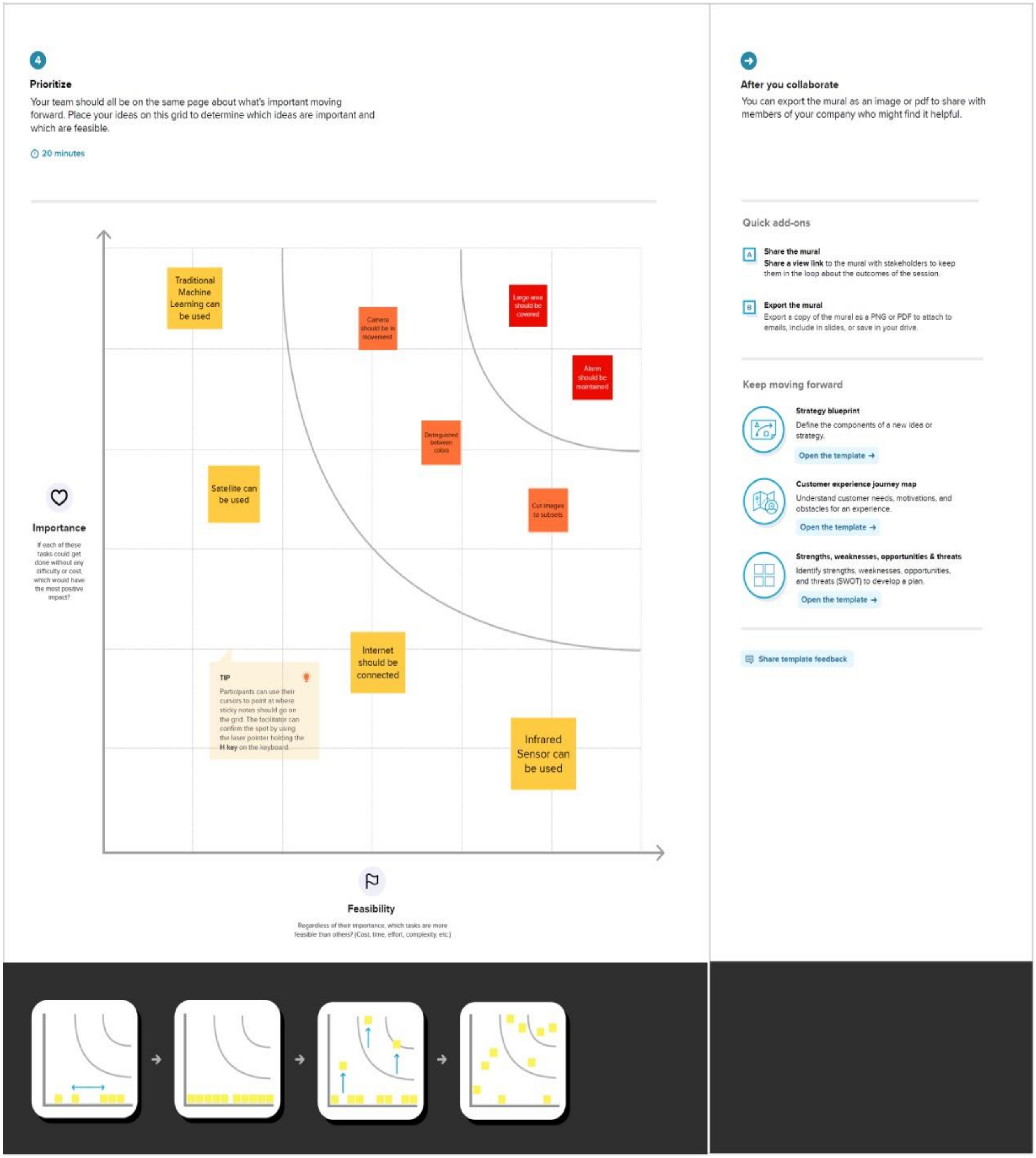
3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

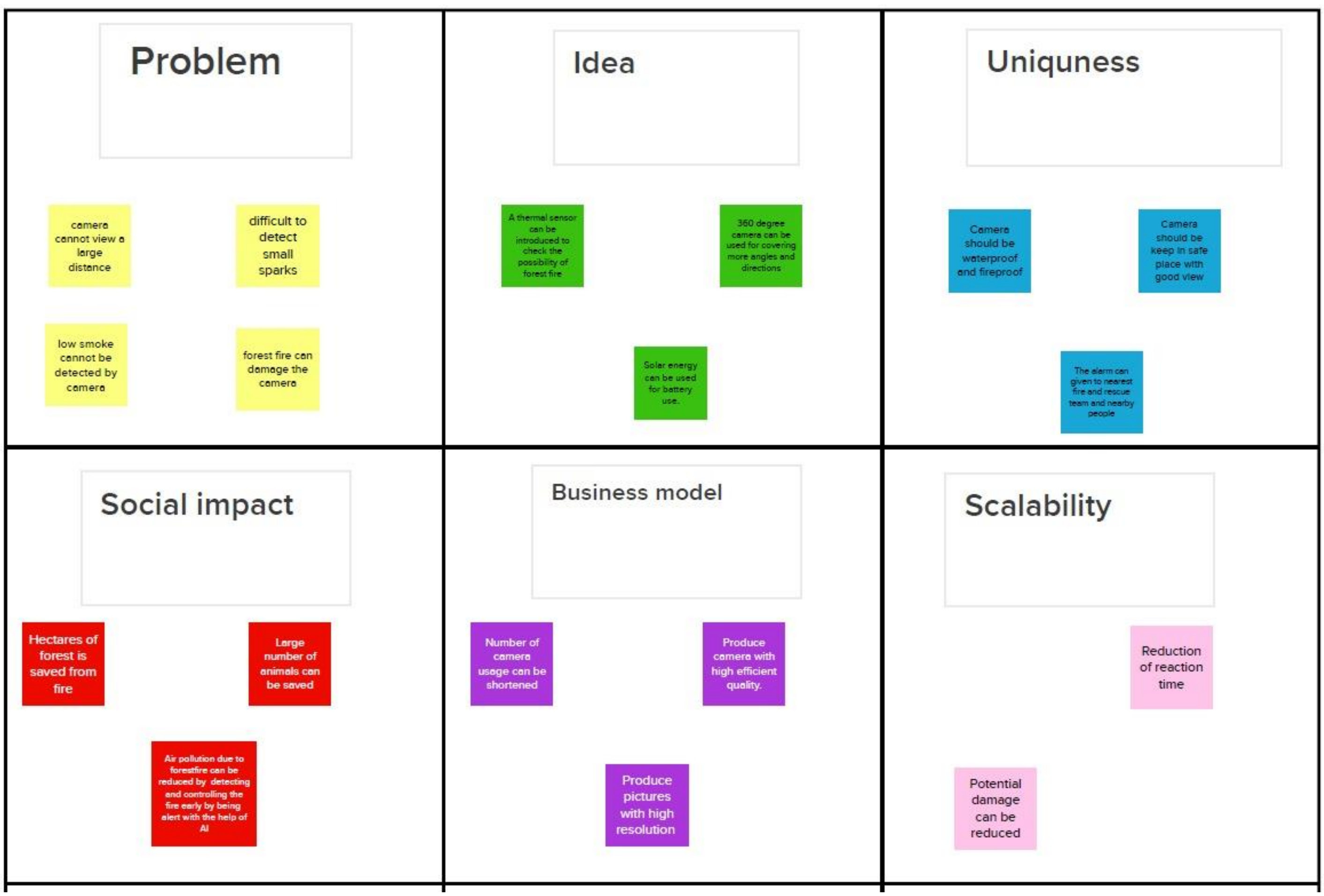


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
SHERJIN RAJ
ADARSH
ANTO SHINE JOVEMON
DESMOND SAMUEL
SHABIN
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Camera should be in movement
Alarm should be maintained
Image should be seen by humans
Smoke detecter should not be used
Small changes should be detected
Latitude and Longitude should be correct
Aerial based
Satellite can be used
Infrared Sensor can be used
Traditional Machine Learning can be used
Internet should be connected
Distinguished between colors
Large area should be covered
Cut images to subsets

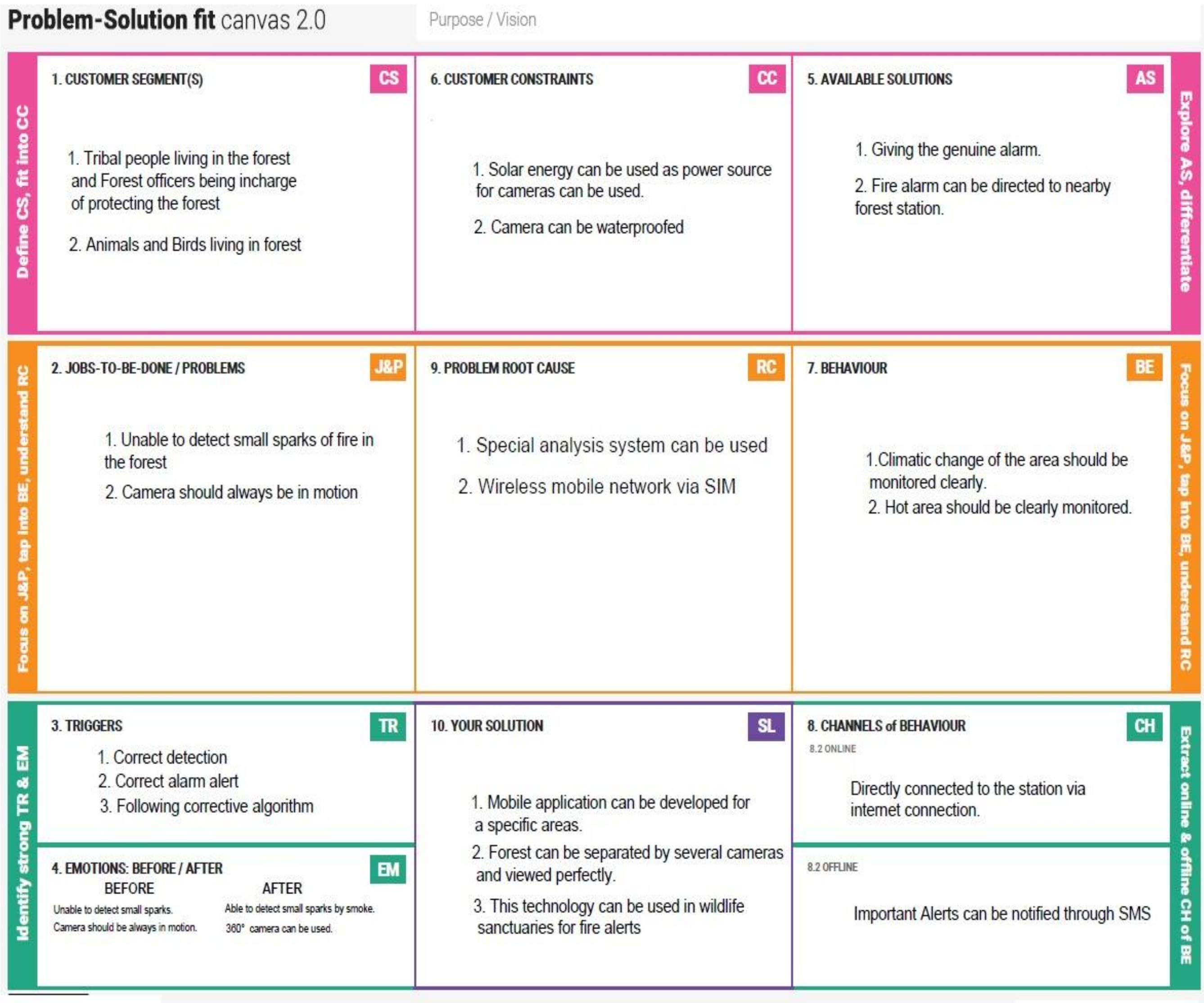

4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Traditional Machine Learning can be used
Camera should be in movement
Large area should be covered
Alarm should be maintained
Distinguished between colors
Satellite can be used
Cut images to subsets
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Internet should be connected
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
Infrared Sensor can be used
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
A
Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
B
Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.
Keep moving forward
Strategy blueprint
Define the components of a new idea or strategy.
Open the template →
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →
Share template feedback

3.3 Proposed Solution

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.



3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Application
FR-2	User Confirmation	Confirmation via Email or message Confirmation via OTP
FR-3	User Alert	The warning to user is provided through Alarm
FR-4	User Connection	Surveillance is provided by Camera or Drone
FR-5	Fire Detection	The camera output is verified by Artificial Intelligence
FR-6	Signal Transmission	The information processed can transferred to destinations by internet and network towers

4.2 Non-functional Requirements:

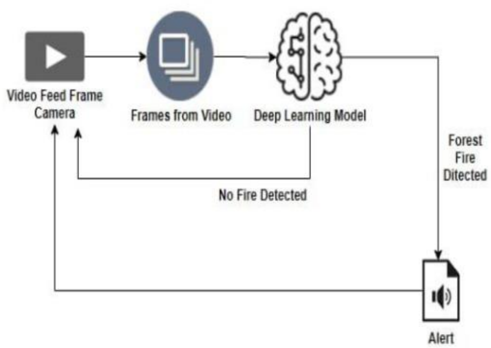
Following are the non-functional requirements of the proposed solution.

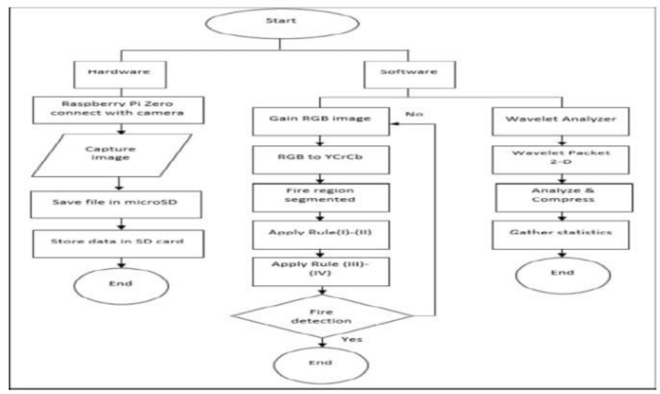
FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
NFR-1	**Usability**	Provides early warning of forest fire to avoid massive forest damage and to protect life in forest

NFR-2	**Security**	Provides protection and security to the tribal people , animals in the forest and also the entire forest
NFR-3	**Reliability**	The system is reliable and trust worthy due to fast and accurate fire detection process via camera using Artificial Intelligence and wireless transmission for signal and alert over areas . The system components and cameras are durable that can mostly survive disaster conditions
NFR-4	**Performance**	The system detects small sparks of fire in a location and identified by Artificial Intelligence with high accuracy and speed
NFR-5	**Availability**	The surveillance provided by system camera are 360 degree and also can be drones watch over the forest 24*7 (all the time). The power source for camera and transmitting components can be taken from solar energy , so they don’t run out of battery.
NFR-6	**Scalability**	A large area of the forest can be covered under surveillance by using Drones and 360 degree cameras.

5. PROJECT DESIGN

5.1 Data Flow Diagrams





5.2 Solution and Technical Architecture

Solution Architecture

1. ***This Solution Architecture involves four stages.***
   a. ***Input Image***
   b. ***Region Proposal***
   c. ***Feature extraction &classification***
   d. ***Output detection result***

Step 1: We get the input image and discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are

mapped out.

Step 2: The second part of this step will involve the Rectified Linear Unit or ReLU. We will cover ReLU layers and explore how linearity functions in the context of Convolutional Neural Networks.
Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.

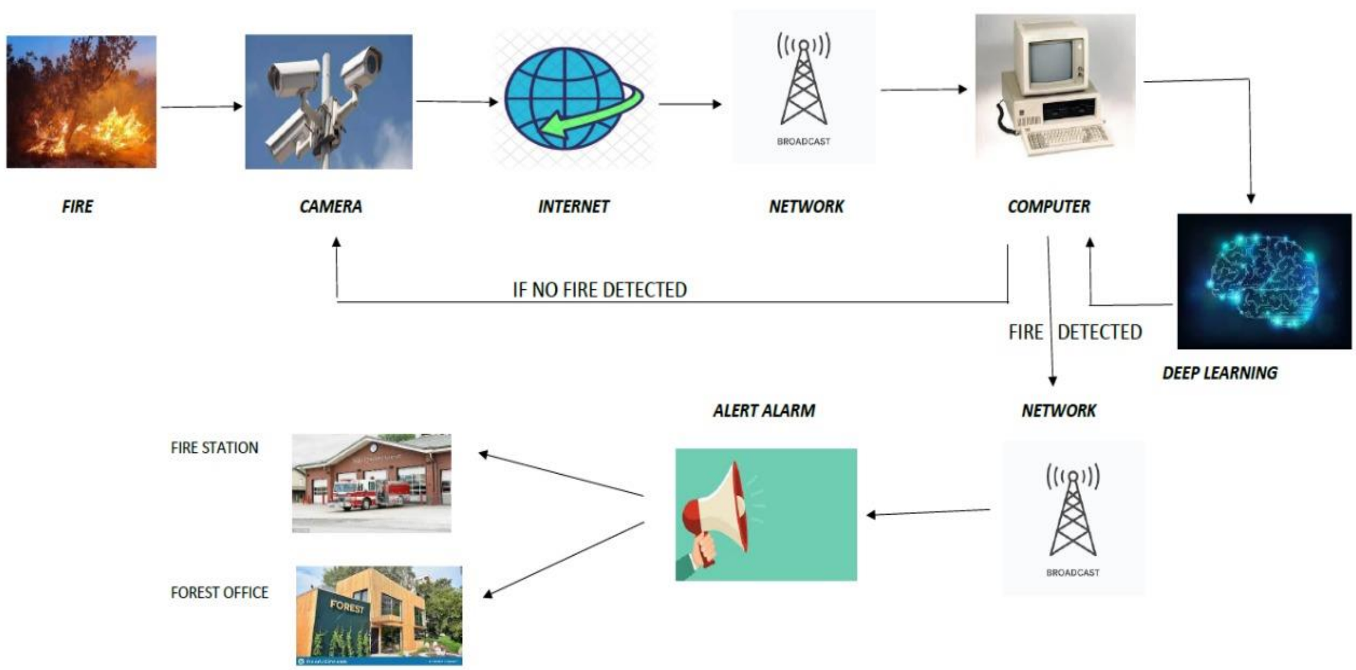
Step 3-Pooling: In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

Step 4 -Flattening: This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional NeuralNetworks.

Step 5-FullConnection: In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how

Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

SOLUTION ARCHITECTURE:



Technology Architecture

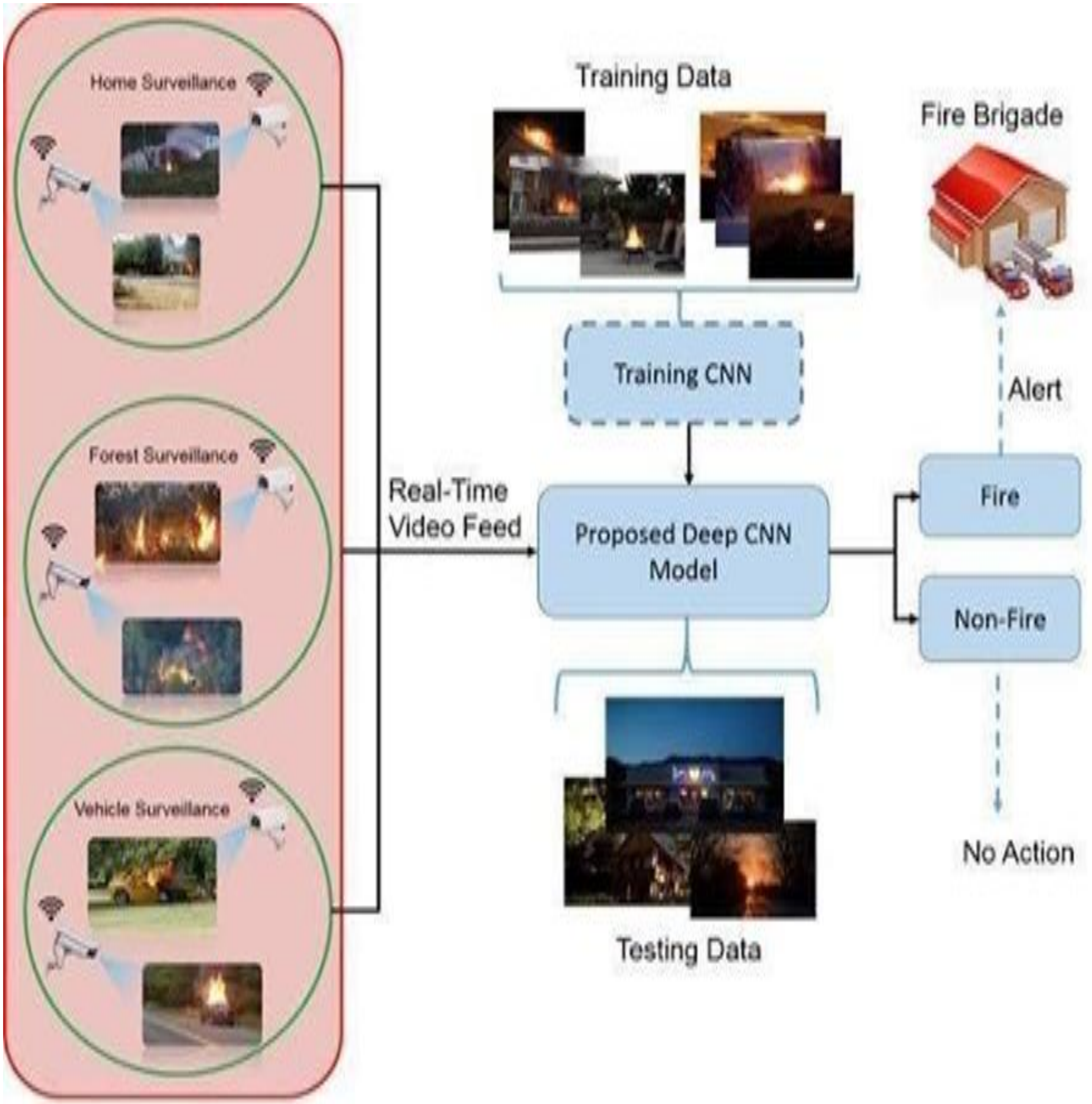


Table 1: Components and Technologies

S.No	***Component***	***Description***	***Technology***
1.	User Interface (Camera)	How user interacts or see the video feed into the computer	High quality camera
2.	Camera and Drones	Watching the Forest ; surveillance provided (24*7) all the time	Pan tilt zoom cameras can be used
3.	Fire System	Identifying smoke by clustering motions with a time input to reduce the number of false alarm	Ura Fire System
4.	Communication	To send the videos from camera to the system	Network Tower
5.	Cloud Database	Database Service on Cloud	IBM Cloud
6.	Application to get the video feed	It gets the image and helps the CNN so check whether fire is present	IBM Watson assistant

7.	Sensor	Rotates the camera 360 degree every 4 to 6 minutes in a day OSS at the tower has a wireless connection to the user computer	Optical Sensor can be used
8.	Image recognizer	It learn and extract complex image features effectively	CNN algorithms can be used
9.	Detector	It will send an alert sound if the CNN detects the fire	
10.	CNN	Gets the image Process it and finds whether fire occurs or not	Four algorithms are used Faster-RCNN , R-FCN , SDD , YOLO V3

Table 2 : Application Characteristics

s. no	***Characteristics***	***Description***	***Technology***
1.	Open CV	Open – Source Library for image processing	PYTHON PROGRAMING LANGUAGE

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Forest Office Control Room (Normal times)	Surveillance	USN-1	The camera watches the activities in forest area and feed the video to forest office control room	The user gets the knowledge what is happening in forest and be alert	Medium	Sprint-2
Fire Department	Being Alerted by Alarm	USN-2	As a user, they will be alerted earlier or at the starting stage of Forest Fire and location can be shared	Due to the early alert received , the fire department can arrive to the fire spot early and control forest fire easily	High	Sprint-1
Police Patrol Control Room	Being Alerted by Alarm	USN-3	As a user, they will be alerted earlier or at starting stage of Forest Fire with location	Due to the early alert received, the police patrol can arrive to fire spot earlier and help to evacuate nearby villages , save tribal peoples , sealing certain areas to prevent the unwanted entry of outsiders into danger zone	High	Sprint-1
Forest Office Control Room (in case of Forest Fire Alert)	Being Alerted by Alarm	USN-4	As a user, they will be alerted earlier or at starting stage of Forest Fire with location	Due to the early alert received, the forest department can arrive earlier to fire spot and protect nearby tribal people, nearby villagers and also animals as possible	High	Sprint-1

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	DATA COLLECTION	USN-1	Data collected by sensors aboard orbiting satellites, carried aboard aircraft, or installed on the ground provide a wealth of data that can be used to assess conditions before a burn and track the movement of a wildfire in near real-time.	10	High	• Shabin P U • Sherjin Raj S • Adarsh J • Desmond Samuel J • Anto Shine Jovemon R M
Sprint-1	IMAGE PREPROCESSING	USN-2	Image processing-Image processing technique automatically detect forest fires around the world by using infrared(IR) images sourced from satellites and CNN used for image recognition and tasks that involve the processing of pixel data.	7	Medium	• Shabin P U • Sherjin Raj S • Adarsh J • Desmond Samuel J • Anto Shine Jovemon R M
Sprint-2	TRAINING AND TESTING	USN-3	The model is trained for detecting the fire by training with real time work and the testing is done according the accuracy of the model	10	high	• Shabin P U • Sherjin Raj S • Adarsh J • Desmond Samuel J • Anto Shine Jovemon R M

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	REVIEWING THE MODEL	USN-4	The main task is to check that the model is efficient to work in real time to ensure there is no error in the model	7	Medium	• Shabin P U • Sherjin Raj S • Adarsh J • Desmond Samuel J • Anto Shine Jovemon R M
Sprint-4	IMPLEMENTATION	USN-5	After completing every step the model is implemented on the forest and the quick responses is collected from forest organization	10	High	• Shabin P U • Sherjin Raj S • Adarsh J • Desmond Samuel J • Anto Shine Jovemon R M

6.2 Sprint Delivery Schedule

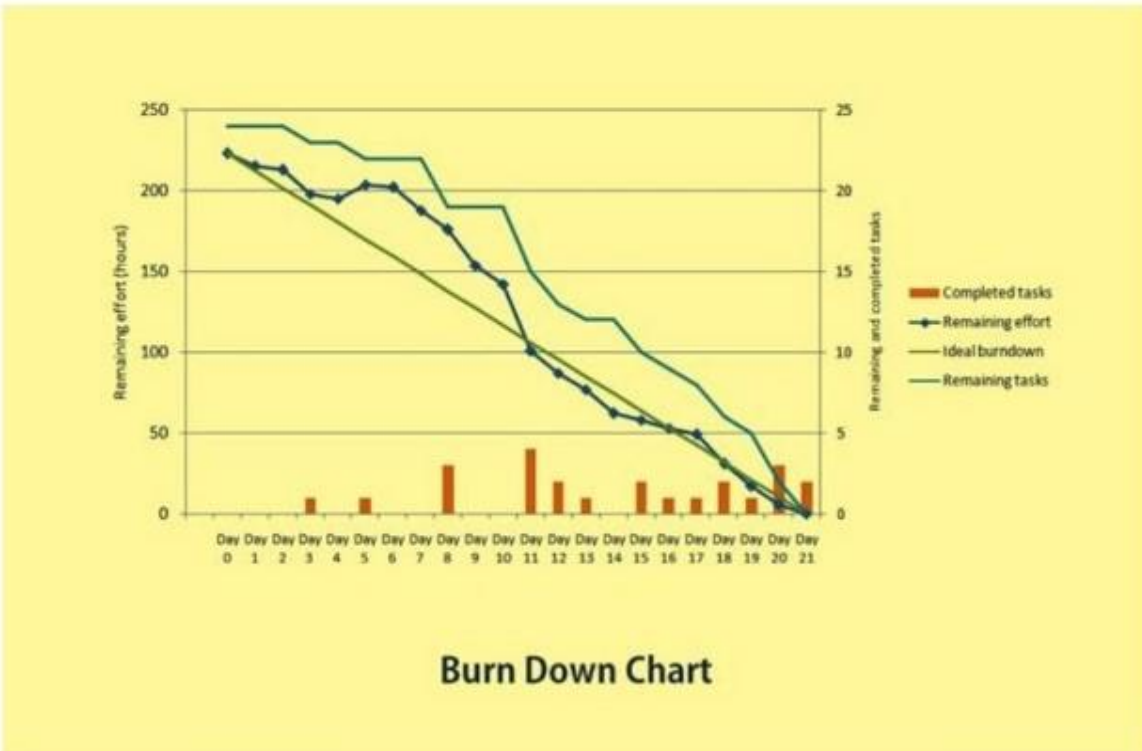
Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	7	08 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	8	15 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	7	20 Nov 2022

Velocity:

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

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



Burn Down Chart

7 CODING AND SOLUTION

7.1 Feature 1

7.1.1 Language used: Python

7.1.2 Tools/IDE:Google Colab

In [107…
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Data Collection

In [108…
```
!unzip '/content/drive/MyDrive/archive.zip'
```

```
Archive:  /content/drive/MyDrive/archive.zip
replace Dataset/Dataset/test_set/forest/0.48007200_1530881924_final_forest.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

Image Pre-processing

In [109…
```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   rotation_range=180,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
train = train_datagen.flow_from_directory('/content/Dataset/Dataset/test_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')
test = train_datagen.flow_from_directory('/content/Dataset/Dataset/train_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')
```

```
Found 121 images belonging to 2 classes.
Found 436 images belonging to 2 classes.
```

Sprint 2

In [110…
```
#Model Building
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPooling2D,Dense,Flatten
import warnings
warnings.filterwarnings('ignore')
```

In [111…

```
#Initializing the model and adding CNN and Dense layers
model = Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=256,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 256)               32514304

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 32,515,457
Trainable params: 32,515,457
Non-trainable params: 0
_________________________________________________________________
```

In [112…

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

In [113…

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

```
Epoch 1/15
 4/14 [=======>......................] - ETA: 27s - loss: 4.0799 - accuracy: 0.5537 - mse: 0.3706
```

```
14/14 [==============================] - 27s 1s/step - loss: 4.0799 - accuracy: 0.5537 - mse: 0.3706 - val_loss: 6.6469 - val_accuracy: 0.6562 - val_ms
e: 0.3400
```

In [114…

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

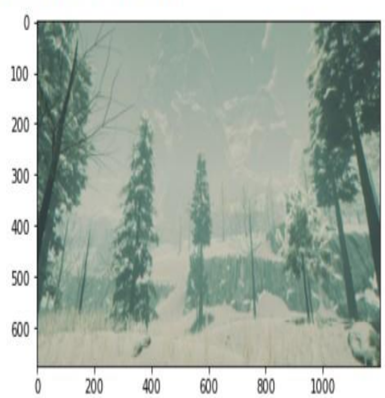
In [116…

```
#Testing the model
from keras.models import load_model
import cv2
import numpy as np
from PIL import Image
from keras.utils import img_to_array
model = load_model('/content/ffd_model.h5')
def prediction(img_path):
    i = cv2.imread(img_path)
    i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
    img = Image.open(img_path)
    img = img.resize((128,128))
    x = img_to_array(img)
    x = np.expand_dims(x,axis=0)
    pred = model.predict(x)
    plt.imshow(i)
    print("%s"%("FOREST FIRE DETECTED! SMS SENT!" if pred==[[1.]] else "NO FOREST FIRE DETECTED"))
```

In [117…

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

```
1/1 [==============================] - 0s 182ms/step
NO FOREST FIRE DETECTED
```



In [21]:

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

```
1/1 [==============================] - 0s 34ms/step
FOREST FIRE DETECTED! SMS SENT!
```

In [ ]:

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

```
ffd_model.h5
```

7.2 Feature 2

In [ ]:
```
!pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting twilio
  Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 26.9 MB/s
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.24.
3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3
```

In [ ]:
```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: twilio in /usr/local/lib/python3.7/dist-packages (7.15.3)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.6.0)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.24.
3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
```

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

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7035 sha256=d7b72b86f6c4197d7039d5af31eba588f2ed25db3c55be17e7deb890119ee
b70
```

```
  Stored in directory: /root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0ecb5f9dfcb7b7e22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
```

In [ ]:

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

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

In [30]:

```
#load the saved model
model = load_model(r'/content/forest1.h5')
#define video
video = cv2.VideoCapture('/content/demo.mp4')
#define the features
name = ['forest','with forest']
```

In [32]:

```
account_sid = 'AC6a4a6501c7a1246424e4186d94ece572'
auth_token = '0b9ce781e03e50ff1746aebbd20c49a5'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
      body='Forest fire is detected , stay alert',
      from_='+13466998949',
      to='+918110851267'
    )

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

```
SMafbfa0ed960c78c528bd58c5396aa8c0
Fire Detected
SMS Sent
```

8 TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
OP_RT_001	Functional	Page	Check if user can upload their file	The sensor senses the fire	Sample 1.png	The input image should be uploaded to the application successfully	Working as expected	PASS		R.Pranati P.Monisha
OP_RT_002	Functional	Page	Check if user cannot upload unsupported files	1) The sensor senses the fire 2)checks with the pre-uploads images	installer.exe	The application should not allow user to select a non image file	User is able to upload any file	FAIL	BUG_HP_002	S.Preethi R.Sangeetha Priya
OP_RT_003	Functional	Page	Checks whether the page redirects to the result page to the given output	1) The sensor senses the fire 2)checks with the pre-uploaded images 3)checks if there is fire detection	Sample 1.png	The page should redirect to the results page	Working as expected	PASS		R.Pranati
MB_RT_001	Functional	Backend	Checks if all the routes are working properly	1) The sensor senses the fire 2)checks with the pre-uploaded images 3)checks if there is fire detection	Sample 1.png	All the routes should properly work	Working as expected	PASS		P.Monisha
N_DC_001	Functional	Model	Checks whether the model can handle various image sizes	1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input	Sample 1.png Sample 1 XS.png Sample 1 XL.png	The model should rescale the image and predict the results	Working as expected	PASS		R.Sangeetha Priya
N_DC_002	Functional	Model	Check if the model predicts the digit	1) Open the page 2) Select the input images	Sample 1.png	The model should predict the number	Working as expected	PASS		R.Sangeetha Priya S.Preethi
N_DC_003	Functional	Model	Check if the model can handle complex input image	1) Open the page 2) Select the input images 3) Check the results	Complex Sample.png	The model should predict the number in the compex image	The model fails to identify the digit since the model is not built to handle such data	FAIL	BUG_M_001	R.Pranati P.Monisha
RL_DC_001	Functional	Result Page	Verify the elements	1) Open the page 2) Select the input image 3) Check if all the UI elements are displayed properly	Sample 1.png	The Result page must be displayed properly	Working as expected	PASS		R.Pranati P.Monisha
RL_DC_002	Functional	Result Page	Check if that image is displayed properly	1) Open the page 2) Select the input image 3) Check if the input image are displayed	Sample 1.png	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL	BUG_RP_001	R.Sangeetha Priya S.Preethi
RL_DC_003	Functional	Result Page	Checks whether the displayed prediction is accurate	1) Open the page 2) Select the input image 3) Check if all the other predictions are displayed	Sample 1.png	The other predictions should be displayed properly	Working as expected	PASS		R.Pranati R.Sangeetha Priya

8.2 User Acceptance Testing

1. Purpose of Document

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing are done.

The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. The arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	1	2	0	4
Duplicate	0	0	0	0	0
External	0	0	2	1	3
Fixed	4	2	4	1	11
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won't Fix	0	0	0	1	1
Totals	5	3	9	4	21

3. Test Case Analysis

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

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

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/

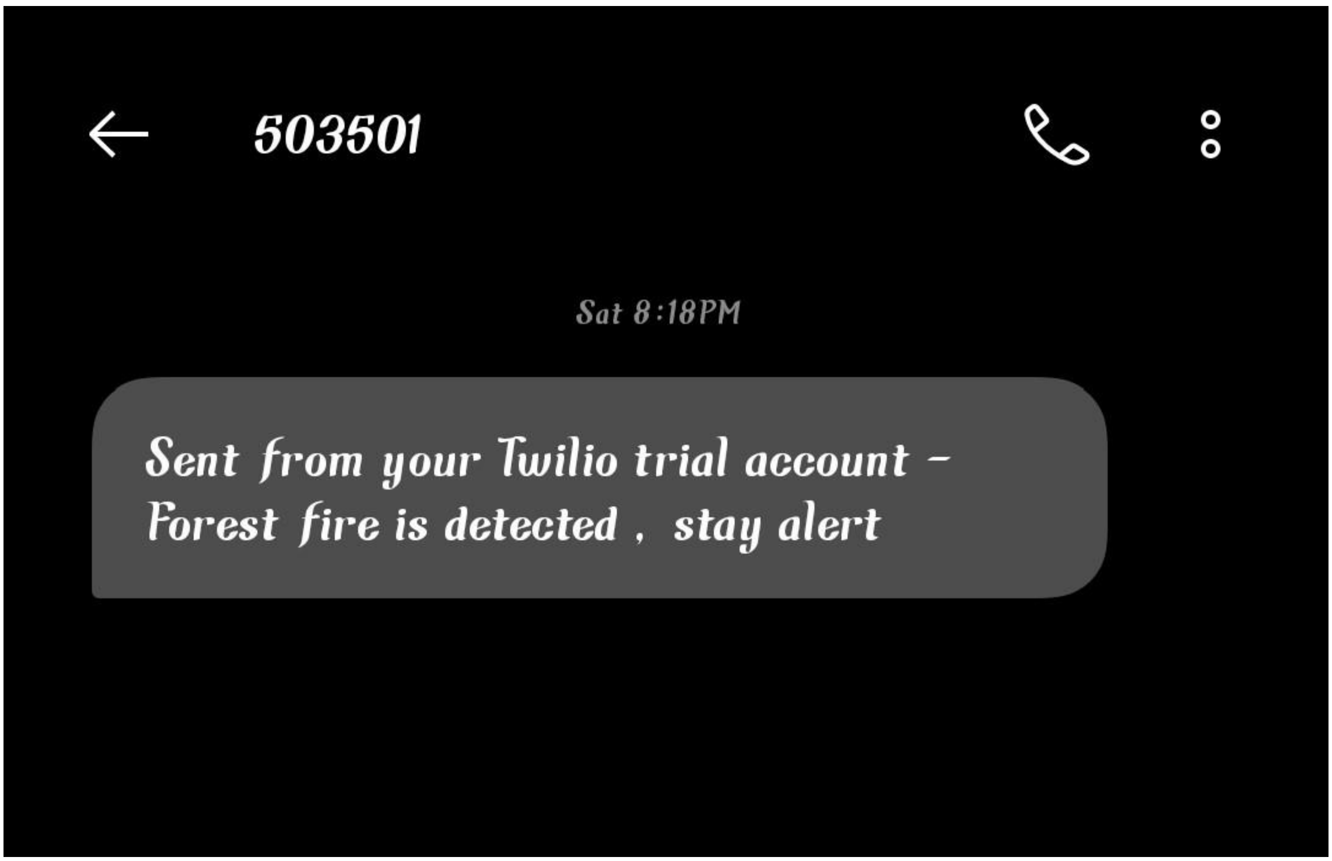
Script: locust.py

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**



503501
Sat 8:18PM
Sent from your Twilio trial account -
Forest fire is detected , stay alert

10 ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. ***The proposed model can be used in combination with a nightcamera and a thermal camera in a forest to identify tiny fire signs.***
2. ***More datasets and images can be used to train for a more accurateoutcome when detecting fiame destruction ability.***
3. ***The model can be implemented in mobile***
4. ***applications for camping experience enthusiasts.***

DISADVANTAGES:

1. ***The model works for limited information.***
2. ***The accuracy is low because to the limited quantity/quality of photos in the dataset, but this may easily be increased by changing the dataset.***
3. ***The small amount of fire amount detection can also cause to trigger the alarm.***

APPLICATIONS:

1. ***It will contribute to surveillance technology that improves the accuracy and predictability of fire detection.***
2. ***able to detect the fire forest more precisely, as well as some forest plants and wildlife.***
3. ***Detect the amount of dangers that should be treated and those that should not. extra assistance in contacting fire fighters for assistance system.***

11 CONCLUSION

Forest fires are a major cause of rain forest and savanna degradation. This model will aid in minimizing destruction by anticipating it to the system, allowing individuals to react more quickly and prevent it. The proposed methodology would deconstruct the threat to the environment by converting the image collected into signals that will trigger an alarm. This system transmits video images to a model,which recognizes them and determines whether to send a threat alert or not. The model extracts data from video feeds and defines image

processing into RGB data for signal response modelling.

12 FUTURE SCOPE

SThe availability of fire-fighting technology brings us one step closer to new AI for detection and security in the forest and at home. With the addition of a motion sensor, the technology can simply expand to compact decision-making with the addition of new software and hardware. The system is utilized as a drone and surveillance system UAV to expand the surveillance area and detect heat signatures in order to identify human from fire plasma signatures.

13 APPENDIX

13.1 Source Code

In [86]:

```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: twilio in /usr/local/lib/python3.7/dist-packages (7.15.2)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.28.1)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.6.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.26.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
```

In [87]:

```
pip install playsound
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: playsound in /usr/local/lib/python3.7/dist-packages (1.2.2)
```

In [88]:

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

In [32]:

```
account_sid = 'AC6a4a6501c7a1246424e4186d94ece572'
auth_token = '0b9ce781e03e50ff1746aebbd20c49a5'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
      body='Forest fire is detected , stay alert',
      from_='+13466998949',
      to='+918110851267'
     )

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

```
SMafbfa0ed960c78c528bd58c5396aa8c0
Fire Detected
SMS Sent
```